



**ITS**

Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

**SISTEM PEMETAAN SUHU PERMUKAAN LAHAN  
MENGUNAKAN SENSOR INFRAMERAH UNTUK  
PENDETEKSI DINI KEBAKARAN**

Widya Yanuar Samsudin  
NRP 07111545000006

Dosen Pembimbing  
Dr. Muhammad Rivai, S.T., M.T.  
Ir. Tasripan, M.T.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*# Halaman ini sengaja dikosongkan #*



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 141599**

**LAND SURFACE TEMPERATURE MAPPING SYSTEM  
USING INFRARED SENSOR FOR EARLY DETECTION  
OF FIRE**

Widya Yanuar Samsudin  
NRP 07111545000006

Supervisors  
Dr. Muhammad Rivai, S.T., M.T.  
Ir. Tasripan, M.T.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*# Halaman ini sengaja dikosongkan #*



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Sistem Pemetaan Suhu Permukaan Lahan Menggunakan Sensor Inframerah untuk Pendeteksi Dini Kebakaran” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 16 Januari 2018

Widya Yanuar Samsudin  
NRP. 0711154500006

*# Halaman ini sengaja dikosongkan #*

**SISTEM PEMETAAN SUHU PERMUKAAN LAHAN  
MENGUNAKAN SENSOR INFRAMERAH UNTUK  
PENDETEKSI DINI KEBAKARAN**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Elektronika  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

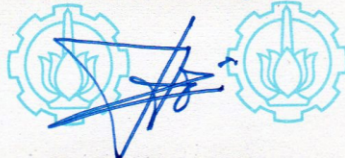
**Dosen Pembimbing I**



**Dr. Muhammad Rivai, S.T., M.T.**

**NIP. 196904261994031003**

**Dosen Pembimbing II**



**Ir. Tasripan, M.T.**

**NIP. 196204181990031004**



*# Halaman ini sengaja dikosongkan #*

# SISTEM PEMETAAN SUHU PERMUKAAN LAHAN MENGUNAKAN SENSOR INFRAMERAH UNTUK PENDETEKSI DINI KEBAKARAN

Nama : Widya Yanuar Samsudin  
Dosen Pembimbing I : Dr. Muhammad Rivai, S.T., M.T.  
Dosen Pembimbing II : Ir. Tasripan, M.T.

## ABSTRAK

Saat ini kebakaran lahan sering terjadi di Indonesia. Pencegahan dapat dilakukan dengan cara mendeteksi titik panas (*hotspot*) sebelum terjadinya kebakaran. Titik panas adalah daerah yang memiliki suhu permukaan relatif lebih tinggi dibandingkan daerah di sekitarnya.

Pada penelitian ini telah dirancang suatu sistem pencegahan yang dapat memetakan suhu suatu permukaan area atau lahan. Pengukur suhu berupa sensor inframerah thermopile akan mengukur intensitas cahaya inframerah yang terpancar dari suatu titik / objek tanpa kontak. Intensitas cahaya inframerah ini proporsional terhadap suhu objek tersebut. Tegangan keluaran dari sensor thermopile dikuatkan oleh rangkaian pengondisi sinyal dan kemudian dibaca *Analog to Digital Converter* pada mikrokontroler Arduino Nano. Sistem pemindai yang menggunakan motor servo dan motor stepper, digunakan untuk mengubah arah titik pengukuran suhu beberapa titik dalam area lahan sehingga suhu suatu area di sekitar sistem dapat dipetakan. Data-data suhu pada area tersebut kemudian divisualisasikan pada perangkat lunak pemetaan komputer yang dirancang menggunakan IDE Microsoft Visual Studio.

Sistem pengukur suhu berdasarkan sensor inframerah thermopile dapat mengukur suhu tanpa kontak dengan rerata kesalahan sebesar 6,4% pada rentang pengukuran suhu 37 °C s.d. 170 °C. Sistem pemindai yang menggunakan motor servo dan stepper dapat mengubah arah titik pengukuran sehingga dapat mengukur dan menggambarkan profil suhu permukaannya dengan perangkat lunak pemetaan.

Kata Kunci: inframerah, kebakaran, suhu, thermopile

*# Halaman ini sengaja dikosongkan #*

## **LAND SURFACE TEMPERATURE MAPPING SYSTEM USING INFRARED SENSOR FOR EARLY DETECTION OF FIRE**

*Name* : Widya Yanuar Samsudin  
*1<sup>st</sup> Supervisor* : Dr. Muhammad Rivai, S.T., M.T.  
*2<sup>nd</sup> Supervisor* : Ir. Tasripan, M.T.

### **ABSTRACT**

*Currently landfires are often occur in Indonesia. Prevention can be done by detecting hotspots before the case of fire. Hotspots are areas that have relatively higher surface temperatures than the surrounding area.*

*In this research has designed a prevention system that can map the temperature of a surface area or land. The thermopile infrared sensor will measure the intensity of the infrared light emitted from a point / object without contact. The intensity of infrared light is proportional to the temperature of the object. The output voltage of the thermopile sensor is amplified by the signal conditioning circuit and then is read by Analog to Digital Converter on the Arduino Nano microcontroller. Scanner system that use servo and stepper motor, are used to change the direction of the point of temperature measurement of multiple points within the area of the land so that the temperature of an area around the system can be mapped. Temperature data in the area is then visualized in computer mapping software that is designed using Microsoft Visual Studio IDE.*

*The temperature measuring system based on the thermopile infrared sensor can measure the temperature without contact with a mean error of 6.4% over a measurement range of 37 °C - 170°C. Scanner system that use servo and stepper motors can change the direction of measurement points, so that is able to measure and describe their surface temperature profiles with mapping software.*

*Keywords: fire, infrared, temperature, thermopile*

*# Halaman ini sengaja dikosongkan #*



## KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa serta tidak lupa doa semoga tetap tercurah berkah dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul:

**SISTEM PEMETAAN SUHU PERMUKAAN LAHAN  
MENGUNAKAN SENSOR INFRAMERAH UNTUK  
PENDETEKSI DINI KEBAKARAN.**

Selama pelaksanaan penelitian ini, penulis mendapatkan bantuan dari berbagai pihak. Penulis sampaikan rasa terima kasih kepada berbagai pihak yang mendukung dan membantu dalam ini, diantaranya:

1. Dr. Muhammad Rivai, S.T., M.T. dan Ir. Tasripan, M.T. selaku dosen pembimbing, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian ini.
2. Dr. Ir. Hendra Kusuma, M.Eng.Sc., Ir. Harris Pirngadi, M.T., Dr.Eng. Ir. Totok Mujiono, M.IKom., dan Dr. Astria Nur Irfansyah, S.T., M.Eng. selaku dosen penguji Sidang Ujian Tugas Akhir.
3. Dr. Eng. Ardyono Priyadi, S.T., M.Eng. selaku Kepala Departemen Teknik Elektro ITS Surabaya.
4. Suwito, S.T., M.T. selaku dosen wali yang memberikan arahan selama masa studi.
5. Seluruh dosen bidang studi elektronika dan seluruh dosen Teknik Elektro ITS.
6. Kedua orang tua tercinta, Bapak Samsudin dan Ibu Rukijah, yang memberikan dukungan secara materiil dan moril.
7. Seluruh rekan-rekan bidang studi elektronika lintas jalur 2015 dan laboratorium B402, serta teman-teman yang turut membantu dalam penyelesaian penelitian ini.

Penulis sadar bahwa penelitian ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 16 Januari 2018

Penulis

*# Halaman ini sengaja dikosongkan #*

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>PERNYATAAN KEASLIAN.....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>21</b>
1.1. LATAR BELAKANG .....	21
1.2. PERUMUSAN MASALAH.....	22
1.3. TUJUAN PENELITIAN .....	22
1.4. BATASAN MASALAH .....	23
1.5. METODOLOGI PENELITIAN .....	23
1.6. SISTEMATIKA PENULISAN .....	24
1.7. RELEVANSI.....	24
<b>BAB II DASAR TEORI.....</b>	<b>27</b>
2.1. KEBAKARAN LAHAN .....	27
2.1.1. Titik Panas ( <i>Hotspot</i> ) .....	27
2.1.2. Lahan Gambut.....	27
2.2. TERMOMETER INFRAMERAH .....	29
2.3. SENSOR RADIASI INFRAMERAH .....	30
2.3.1. Thermopile .....	33
2.4. THERMISTOR .....	35
2.4.1. Instrumentasi Thermistor .....	36
2.5. RANGKAIAN <i>AMPLIFIER</i> .....	38
2.5.1. Dasar <i>Operational Amplifier</i> .....	38
2.5.2. <i>Non-Inverting Voltage Follower</i> .....	40
2.5.3. Penguat Diferensial .....	41

2.6. MOTOR SERVO.....	43
2.7. MOTOR STEPPER.....	44
2.8. SENSOR MAGNETOMETER.....	47
2.9. SENSOR AKSELEROMETER .....	49
2.10. MIKROKONTROLER .....	50
2.11. MICROSOFT VISUAL STUDIO .....	53

### **BAB III PERANCANGAN SISTEM .....55**

3.1. BLOK DIAGRAM.....	55
3.2. RANGKAIAN REGULATOR CATU DAYA.....	57
3.3. RANGKAIAN BOARD MIKROKONTROLER ARDUINO.....	57
3.4. RANGKAIAN SENSOR MAGNETOMETER .....	60
3.4.1. Perangkat Keras Sensor Magnetometer .....	61
3.4.2. Perangkat Lunak Sensor Magnetometer .....	62
3.5. RANGKAIAN SENSOR AKSELEROMETER.....	63
3.5.1. Perangkat Keras Sensor Akselerometer .....	65
3.5.2. Perangkat Lunak Sensor Akselerometer .....	66
3.6. PERANCANGAN RANGKAIAN SENSOR INFRAMERAH .....	67
3.6.1. Perangkat Keras Sensor Inframerah.....	69
3.6.2. Perangkat Lunak Sensor Inframerah.....	71
3.7. PERANCANGAN PENGGERAK MOTOR STEPPER .....	75
3.7.1. Perangkat Keras Penggerak Motor Stepper .....	76
3.7.2. Perangkat Lunak Penggerak Motor Stepper .....	76
3.8. PERANCANGAN PENGGERAK MOTOR SERVO .....	77
3.8.1. Perangkat Keras Penggerak Motor Servo .....	78
3.8.2. Perangkat Lunak Penggerak Motor Servo .....	78
3.9. PERANGKAT LUNAK PEMETAAN SUHU .....	78
3.9.1. Fasilitas Port Serial .....	80
3.9.2. Konversi Data Suhu ke Data Piksel .....	80
3.9.3. Menghasilkan Berkas Gambar dari Data Piksel.....	81
3.10. APLIKASI PRAKTIS SISTEM PEMETAAN .....	82

### **BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM.....85**

4.1. PENGUJIAN RANGKAIAN SENSOR THERMOPILE ZTP-135SR .....	85
4.1.1. Pengujian Kalibrasi untuk Mendapatkan $K_p$ .....	86
4.1.2. Pengujian Pengukuran Suhu Objek .....	88
4.2. PENGUJIAN SENSOR AKSELEROMETER DAN MOTOR SERVO .....	91
4.2.1. Pengujian Sensor Akselerometer .....	91
4.2.2. Pengujian Motor Servo .....	93

4.3. PENGUJIAN SENSOR MAGNETOMETER DAN MOTOR STEPPER .....	95
4.3.1. Pengujian Sensor Magnetometer.....	95
4.3.2. Pengujian Motor Stepper.....	97
4.4. PENGUJIAN PERANGKAT LUNAK PEMETAAN.....	100
4.4.1. Pengujian Koordinat Piksel dan Data.....	100
4.4.2. Pengujian Radius pada Pola .....	100
4.5. PENGUJIAN SISTEM .....	101
<b>BAB V PENUTUP.....</b>	<b>107</b>
5.1. KESIMPULAN .....	107
5.2. SARAN.....	107
<b>DAFTAR PUSTAKA .....</b>	<b>109</b>
<b>LAMPIRAN.....</b>	<b>113</b>
<b>BIODATA PENULIS.....</b>	<b>119</b>

*# Halaman ini sengaja dikosongkan #*

## DAFTAR GAMBAR

<b>Gambar 2.1.</b> Lahan Gambut yang Ada di Jambi .....	28
<b>Gambar 2.2.</b> Pancaran Inframerah .....	29
<b>Gambar 2.3.</b> Pertukaran Radiasi Panas .....	31
<b>Gambar 2.4.</b> Diagram Skematik Thermopile .....	34
<b>Gambar 2.5.</b> Penampang Melintang Sensor Thermopile .....	34
<b>Gambar 2.6.</b> Rangkaian Instrumentasi Thermistor .....	38
<b>Gambar 2.7.</b> Rangkaian Ekuivalen Op-Amp Ideal .....	39
<b>Gambar 2.8.</b> Konfigurasi <i>Non-Inverting Voltage Follower</i> .....	41
<b>Gambar 2.9.</b> Konfigurasi Rangkaian Penguat Diferensial .....	42
<b>Gambar 2.10.</b> Blok Diagram Rangkaian Kontrol Motor Servo .....	43
<b>Gambar 2.11.</b> Sinyal PWM dan Posisi Angular Servo .....	44
<b>Gambar 2.12.</b> Contoh Motor Stepper dengan 4 <i>Coils</i> Stator .....	45
<b>Gambar 2.13.</b> Metode <i>Wave Drive (Full Step)</i> .....	46
<b>Gambar 2.14.</b> Metode 2 <i>Phases On (Full Step)</i> .....	46
<b>Gambar 2.15.</b> Metode <i>Half Step</i> .....	46
<b>Gambar 2.16.</b> Metode <i>Microstep</i> .....	46
<b>Gambar 2.17.</b> Ilustrasi Medan Magnet Bumi .....	48
<b>Gambar 2.18.</b> Konstruksi Akselerometer VC .....	50
<b>Gambar 2.19.</b> Arsitektur Mikrokontroler .....	51
<b>Gambar 3.1.</b> Blok Diagram Sistem.....	55
<b>Gambar 3.2.</b> Ilustrasi Arah Putaran Motor (a) Stepper, (b) Servo .....	56
<b>Gambar 3.3.</b> Rangkaian Regulator Catu Daya.....	56
<b>Gambar 3.4.</b> Bentuk Fisik Arduino Nano .....	57
<b>Gambar 3.5.</b> Pinout dari Arduino Nano.....	58
<b>Gambar 3.6.</b> Diagram Alir Program pada Arduino Nano .....	59
<b>Gambar 3.7.</b> Skematik Modul GY-282.....	60
<b>Gambar 3.8.</b> Bentuk Fisik Modul GY-282 .....	61
<b>Gambar 3.9.</b> Hubungan Arduino Nano dengan Modul GY-282.....	62
<b>Gambar 3.10.</b> Orientasi Sumbu Sensitivitas dan Polaritas MPU6050 .....	63
<b>Gambar 3.11.</b> Bentuk Fisik Modul GY-521 .....	64
<b>Gambar 3.12.</b> Skematik Modul GY-521.....	65
<b>Gambar 3.13.</b> Hubungan Arduino Nano dengan Modul GY-521.....	65
<b>Gambar 3.14.</b> Bentuk Fisik Thermopile ZTP-135SR .....	67
<b>Gambar 3.15.</b> Fungsi Nilai Resistansi Thermistor dengan Suhu .....	68
<b>Gambar 3.16.</b> Rangkaian Pengondisi Sinyal Thermopile .....	69
<b>Gambar 3.17.</b> Rangkaian Pengondisi Sinyal Thermistor.....	71
<b>Gambar 3.18.</b> Bentuk Fisik Motor Stepper 28BYJ-48 5V.....	75

<b>Gambar 3.19.</b> Bentuk Fisik ULN2003A .....	76
<b>Gambar 3.20.</b> Skematik Rangkaian Penggerak Motor Stepper .....	77
<b>Gambar 3.21.</b> Bentuk Fisik Motor Servo MG995 .....	77
<b>Gambar 3.22.</b> Skematik Rangkaian Penggerak Motor Servo .....	78
<b>Gambar 3.23.</b> Diagram Alir Pemrograman Pemetaan .....	79
<b>Gambar 3.24.</b> Aplikasi Praktis Sistem Pemetaan .....	82
<b>Gambar 4.1.</b> Bentuk Fisik dari Sistem .....	85
<b>Gambar 4.2.</b> Pengujian Sensor Thermopile.....	86
<b>Gambar 4.3.</b> Nilai Faktor Kalibrasi Terhadap Suhu Objek Pengujian 88	
<b>Gambar 4.4.</b> Persentase <i>Error</i> pada Pengukuran Suhu Objek.....	89
<b>Gambar 4.5.</b> Pengujian pada Objek Bersuhu Konstan .....	91
<b>Gambar 4.6.</b> Pengujian Sensor Akselerometer.....	92
<b>Gambar 4.7.</b> <i>Error</i> Pengukuran Sensor Akselerometer.....	93
<b>Gambar 4.8.</b> Ilustrasi (a) Kemiringan dan (b) Putaran Servo .....	94
<b>Gambar 4.8.</b> Perbandingan Kemiringan Terukur dengan Servo.....	95
<b>Gambar 4.9.</b> Pengujian Sensor Magnetometer dengan Arah 90° .....	95
<b>Gambar 4.10.</b> Pengujian Sensor Magnetometer dengan Arah 0° .....	96
<b>Gambar 4.11.</b> <i>Error</i> pada Pengujian Sensor Magnetometer .....	97
<b>Gambar 4.12.</b> Pengujian Motor Stepper dengan <i>Set-Point</i> 0° .....	99
<b>Gambar 4.13.</b> Pengujian Koordinat Piksel dan Data .....	100
<b>Gambar 4.14.</b> Pengujian Radius pada Pola .....	101
<b>Gambar 4.15.</b> Lingkungan Pengujian Sistem .....	102
<b>Gambar 4.16.</b> Hasil Pengujian Sistem.....	102
<b>Gambar 4.17.</b> Lingkungan Pengujian Ke-2 Sistem .....	103
<b>Gambar 4.18.</b> Hasil Pengujian Ke-2 Sistem .....	103
<b>Gambar 4.19.</b> Lingkungan Pengujian Ke-3 Sistem .....	104
<b>Gambar 4.20.</b> Hasil Pengujian Ke-3 Sistem.....	105



## DAFTAR TABEL

<b>Tabel 2.1.</b> Karakteristik Pertanian dan Kimia dari Gambut .....	28
<b>Tabel 2.2.</b> Penggunaan Praktis Thermistor Model Steinhart-Hart .....	35
<b>Tabel 3.1.</b> Fungsi Pin I/O pada Arduino dalam Sistem .....	58
<b>Tabel 3.2.</b> Hubungan Arduino Nano dengan Modul GY-282 .....	61
<b>Tabel 3.3.</b> Hubungan Arduino Nano dengan Modul GY-521 .....	66
<b>Tabel 3.4.</b> Hubungan Nilai Resistansi Thermistor dengan Suhu .....	68
<b>Tabel 3.5.</b> Aplikasi Praktis Sistem Pemetaan .....	83
<b>Tabel 4.1.</b> Pengujian Untuk Mendapatkan Faktor Kalibrasi .....	87
<b>Tabel 4.2.</b> Pengujian Pengukuran Suhu Objek .....	89
<b>Tabel 4.3.</b> Pengujian pada Objek Bersuhu Konstan .....	90
<b>Tabel 4.4.</b> Pengujian Sensor Akselerometer.....	92
<b>Tabel 4.5.</b> Perbandingan Kemiringan Terukur Dengan Servo.....	94
<b>Tabel 4.6.</b> Hasil Pengujian Sensor Magnetometer.....	96
<b>Tabel 4.7.</b> Pengujian Motor Stepper dengan <i>Set-Point</i> 0° .....	98
<b>Tabel 4.8.</b> Pengujian Motor Stepper dengan <i>Set-Point</i> 20° .....	99
<b>Tabel 4.9.</b> Pengujian Motor Stepper dengan <i>Set-Point</i> 40° .....	99
<b>Tabel 4.10</b> Pengujian Motor Stepper dengan <i>Set-Point</i> 60° .....	99

*# Halaman ini sengaja dikosongkan #*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Kebakaran lahan gambut dan hutan merupakan fenomena alam yang sangat sering terjadi di Indonesia. Hutan yang seharusnya sebagai paru-paru dunia berubah menjadi bencana yang dapat merugikan berbagai pihak. Mulai dari keanekaragaman hayati yang berkurang, habitat hewan-hewan yang hilang, hingga manusia yang dirugikan baik materil maupun non materil akibat kebakaran yang terjadi. Pada 2015, kebakaran terjadi pada 2,6 hektar lahan, yaitu setara dengan ukuran empat setengah kali lipat Pulau Bali dan kerugian bagi negara Indonesia mencapai 221 triliun rupiah [1]. Umumnya kebakaran di Indonesia pada lahan gambut dan hutan. Lahan gambut adalah lahan basah dengan lapisan tanah berair yang terdiri dari bahan tanaman mati dan membusuk [2].

Kebakaran hutan ini disebabkan oleh berbagai macam mulai dari *human error* yaitu untuk pembukaan lahan, puntung rokok, dan pembakaran sampah yang dapat memicu kebakaran yang besar [3]. Selain itu ada juga yang diakibatkan oleh faktor alam yaitu suhu yang tinggi, curah hujan yang rendah, dan bahan yang ada pada lahan yang mudah terbakar. Siklus terjadinya kebakaran hutan terjadi saat musim kemarau yang memang merupakan saat di mana lahan mulai mengering dan suhu sangat tinggi. Sehingga dapat disimpulkan bahwa masalah utama terjadinya kebakaran hutan adalah akibat suhu yang tinggi, kelembaban udara yang rendah dan kurangnya curah hujan di suatu daerah.

Untuk itu diperlukan suatu pencegahan kebakaran lahan. Pencegahan kebakaran lahan diperlukan karena penanganan setelah kebakaran tergolong lebih sulit dan memakan biaya yang lebih banyak. Selain itu, penanganan setelah terjadinya kebakaran akan menimbulkan kerugian materil dan non materil lebih banyak. Jika sudah telanjur terbakar, api di lahan misal gambut jadi sulit dipadamkan karena lahan gambut akan sangat kering sampai kedalaman tertentu [4].

Pencegahan dapat dilakukan dengan pembacaan salah satu parameter penyebab kebakaran hutan yaitu titik panas (*hotspot*). *Hotspot* secara definisi dapat diartikan sebagai daerah yang memiliki suhu permukaan relatif lebih tinggi dibandingkan daerah di sekitarnya. Semakin banyak titik *hotspot*, semakin banyak pula potensi kejadian kebakaran lahan di suatu wilayah [5]. Hasil pemetaan suhu permukaan

lahan dapat digunakan untuk sistem pencegahan kebakaran hutan pada tahap selanjutnya.

Teknologi yang sudah umum untuk pembacaan suhu di permukaan bumi adalah satelit. Salah satu satelit yang digunakan adalah NOAA-AVHRR (*National Oceanic and Atmospheric Administration - Advanced Very-High-Resolution Radiometer*). Satelit ini dapat memberikan data global dua kali dalam sehari [6]. Resolusi tertinggi tanah yang dapat diperoleh dari satelit ini adalah 1,1 km, yang berarti bahwa satelit mencatat informasi terpisah untuk area yang berada di tanah 1,1 x 1,1 km [7]. Unit terekam terkecil ini disebut piksel. Hal-hal tersebut membuat data yang didapat jauh dari *real-time* dan beresolusi yang masih terlalu besar.

Untuk pencegahan kebakaran lahan yang lebih efektif, maka diperlukan alat yang mempunyai spesifikasi yang lebih baik dari teknologi yang sudah ada. Dalam hal ini mempunyai waktu pembacaan yang lebih cepat (mendekati *real-time*) dan resolusi yang lebih kecil.

## **1.2. Perumusan Masalah**

Masalah yang dibahas dalam usulan penelitian ini adalah:

1. Jenis sensor inframerah yang digunakan untuk mengukur suhu permukaan lahan.
2. Bagaimana proses pemindaian suhu permukaan pada suatu area di sekitar sistem?
3. Bagaimana proses visualisasi data-data pengukuran menjadi sebuah *file* gambar pemetaan suhu permukaan lahan?

## **1.3. Tujuan Penelitian**

Tujuan dari penelitian ini adalah terbentuknya prototip sistem pemetaan suhu adalah:

1. Sensor inframerah thermopile digunakan untuk mengukur suhu objek.
2. Motor servo dan motor stepper digunakan untuk pengubahan arah pengukuran dari sensor sehingga beberapa titik pada permukaan suatu area di sekitar sistem dapat diukur suhunya.
3. Proses visualisasi data menjadi *file* gambar/grafis pemetaan menggunakan program yang dikembangkan dengan bantuan IDE Microsoft Visual Studio.

#### 1.4. Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

1. Komputasi penghitungan suhu objek sesuai dengan persamaan pada dasar teori.
2. Komputasi penghitungan resistansi thermistor dan kalibrasi bersumber pada *datasheet*.
3. Pengujian dan kalibrasi pengukur suhu menggunakan objek benda dengan sumber panas alternatif.

#### 1.5. Metodologi Penelitian

Dalam penyelesaian penelitian ini digunakan metodologi sebagai berikut:

1. Studi literatur  
Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan penelitian. Dasar teori ini dapat diambil dari buku-buku, jurnal, *proceeding*, dan artikel-artikel di internet.
2. Perancangan Sistem Pemindaian  
Sistem pemindaian terdiri dari motor servo dan motor stepper. Sehingga diperlukan rangkaian *driver* untuk kedua motor tersebut.
3. Perancangan Rangkaian Sensor Thermopile dan Pengondisi Sinyal  
Tegangan keluaran thermopile perlu dikondisikan dengan spesifikasi dari ADC pada mikrokontroler Arduino Nano.
4. Perancangan Perangkat Lunak Pengambil Data  
Perancangan perangkat lunak pengambil data terdiri dari perangkat lunak untuk membaca nilai ADC dari hasil pengukuran suhu permukaan dan penggerakan motor stepper dan motor servo untuk sistem pemindaian. Selain itu, hasil pemindaian suhu dikirimkan melalui komunikasi serial.
5. Perancangan Perangkat Lunak Pemetaan  
Perancangan perangkat lunak pemetaan di PC terdiri dari penerimaan hasil pemindaian suhu yang dikirimkan oleh mikrokontroler Arduino Nano. Serta perancangan penghasil *file* gambar pemetaan suhu pada PC dengan menggunakan program yang dikembangkan dengan bantuan Microsoft Visual Studio.
6. Pengujian dan Perbaikan Sistem  
Pengujian awal dilakukan per bagian sistem, yaitu sistem pemindaian menggunakan motor servo dan motor stepper, rangkaian sensor inframerah thermopile dan rangkaian pengondisi sinyal, dan pemrograman pemetaan.

Pada tahap selanjutnya, dilakukan pengujian sistem total dengan menempatkan sistem di suatu area. Kemudian di sekitar sistem diletakkan objek yang lebih tinggi suhunya dari objek sekitarnya. Lalu sistem dijalankan dan diamati hasil dari pemetaan dan dilakukan perbaikan sehingga hasil lebih optimal.

7. Penulisan Laporan Penelitian

Tahap penulisan laporan penelitian dilakukan beriringan pengerjaan.

8. Penyusunan Jurnal Ilmiah POMITS

POMITS (Publikasi Ilmiah *Online* Mahasiswa ITS) digunakan untuk mempublikasikan hasil penelitian. Jurnal ilmiah disusun berdasarkan *template* yang ada pada situs *web* POMITS.

## 1.6. Sistematika Penulisan

Laporan penelitian ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

- Bab 1 : Pendahuluan  
Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan, dan relevansi.
- Bab 2 : Dasar Teori  
Bab ini menjelaskan tentang dasar-dasar teori yang dibutuhkan dalam pengerjaan penelitian ini, yang meliputi teori dasar inframerah, termopile, rangkaian pengondisi sinyal, motor stepper, dan motor servo.
- Bab 3: Perancangan Sistem  
Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk pemetaan suhu permukaan lahan.
- Bab 4 : Pengujian dan Pembahasan Sistem  
Bab ini menjelaskan tentang hasil yang didapat dari tiap bagian sistem dan subsistem serta hasil evaluasi sistem tersebut.
- Bab 5 : Penutup  
Bab ini menjelaskan tentang kesimpulan meliputi kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

## 1.7. Relevansi

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mengetahui langkah-langkah dalam pengondisian sinyal sensor

inframerah thermopile untuk pengukuran suhu tanpa kontak dengan objek.

2. Mengetahui cara sistem pemindaian untuk mengubah arah titik pengukuran dengan motor servo dan motor stepper
3. Mengetahui perancangan perangkat lunak sistem pemetaan dengan bantuan IDE Microsoft Visual Studio.

*# Halaman ini sengaja dikosongkan #*



## **BAB II**

### **DASAR TEORI**

#### **2.1. Kebakaran Lahan**

##### **2.1.1. Titik Panas (*Hotspot*)**

*Hotspot* secara definisi dapat diartikan sebagai daerah yang memiliki suhu permukaan relatif lebih tinggi dibandingkan daerah di sekitarnya berdasarkan ambang batas temperatur tertentu yang terpantau [5]. Biasanya digunakan sebagai indikator atau kebakaran lahan dan hutan di suatu wilayah, sehingga semakin banyak titik *hotspot*, semakin banyak pula potensi kejadian kebakaran lahan di suatu wilayah. *Hotspot* sangat direkomendasikan untuk kegiatan deteksi dini adanya kebakaran lahan dan hutan karena merupakan indikator adanya kebakaran lahan/hutan di suatu wilayah. Secara umum, potensi api dapat teridentifikasi jika suhu suatu titik melebihi titik ambang batas  $>310$  K. Sedangkan suatu titik teridentifikasi mengandung api aktif jika suhu terukur melebihi ambang batas  $>360$  K [8].

##### **2.1.2. Lahan Gambut**

Lahan gambut adalah lahan basah dengan lapisan tanah berair yang terdiri dari bahan tanaman mati dan membusuk [2] yang karakteristiknya ditunjukkan pada Tabel 2.1. Yang termasuk kedalam lahan gambut diantaranya adalah moor, bog, mires, hutan rawa gambut dan permafrost tundra. Luas lahan gambut secara keseluruhan mencapai setengah dari luas lahan basah di dunia, dan menutupi 3% dari total luas permukaan bumi. Semua lahan gambut dapat dijumpai di berbagai belahan dunia. Contoh bentang alam lahan gambut ada pada Gambar 2.1.

Kerusakan lahan gambut banyak terjadi karena aktivitas manusia, misalnya konversi hutan gambut menjadi lahan pertanian, perkebunan dan kehutanan. Lahan gambut di Indonesia mengalami laju kerusakan tertinggi. Kerusakan lahan gambut diawali dengan proses pembabatan hutan (*land clearing*). Proses selanjutnya adalah pengeringan lahan yang bertujuan untuk mengeluarkan air yang terkandung dalam tanah gambut. Caranya dengan membuat parit atau saluran drainase agar air mengalir keluar. Proses pengeringan ini menyebabkan turunnya permukaan gambut. Sehingga pohon-pohon yang terdapat di permukaan tanah tidak bisa tegak dengan kuat karena akarnya menyembul. Banyak pohon yang roboh di atas gambut yang tidak sehat.

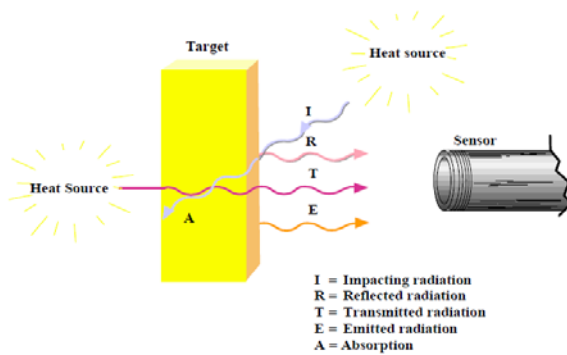


**Gambar 2.1.** Lahan Gambut yang Ada di Jambi [10]

**Tabel 2.1.** Karakteristik Pertanian dan Kimia dari Gambut [9]

Jenis Gambut	Kadar Abu	Kadar Material Organik	pH	Komposisi Kimia				
				N <sub>total</sub>	CaO	P <sub>2</sub> O <sub>5</sub>	K <sub>2</sub> O	Fe <sub>2</sub> O <sub>3</sub>
<i>High-moor</i>	1–5	99–95	2.8–3.6	0.9–2.0	0.1–0.7	0.03–0.2	0.05–0.1	0.03–0.5
<i>Transi-tional</i>	3–8	97–92	3.6–4.8	0.9–3.0	0.5–1.7	0.04–0.3	0.05–0.1	0.1–1.0
<i>Low-Moor</i>	s.d. 12	> 88	4.8–5.8	1.1–3.8	1.2–4.8	0.05–0.4	0.1–0.2	0.2–3.0
	12–20	88–80	4.8–6.6	1.6–3.9	1.2–7.5	0.05–2.0	0.2–0.5	0.1–9.0
	20–50	80–50	4.0–7.0	1.5–3.7	0.3–31.0	0.05–7.5	0.3–0.9	0.2–26.0

Pengeringan pada lahan gambut mempunyai karakteristik tidak dapat kembali (*irreversible*). Sekali air dikeluarkan, gambut akan kehilangan sebagian kemampuannya untuk menyimpan air. Di musim kemarau lahan akan rawan kebakaran. Proses kebakaran lahan gambut merupakan pelepasan karbon dalam jumlah besar ke atmosfer dan memusnahkan keanekaragaman hayati hutan. Sebaliknya di musim hujan hutan tidak bisa menyerap air dengan baik yang menyebabkan bencana banjir [10].



**Gambar 2.2.** Pancaran Inframerah [11]

## 2.2. Termometer Inframerah

Semua benda yang memiliki suhu di atas nol absolut (nol kelvin) memancarkan radiasi inframerah sesuai pada suhunya [11]. Ini disebut dengan radiasi karakteristik. Hal ini disebabkan oleh pergerakan mekanis internal dari molekul. Intensitas dari pergerakan ini bergantung pada suhu dari objek. Karena pergerakan molekul merepresentasikan pemindahan muatan, radiasi elektromagnetik (partikel foton) dipancarkan. Foton-foton ini bergerak pada kecepatan cahaya dan berperilaku sesuai dengan prinsip-prinsip optik yang sudah dikenal. Mereka dapat dibelokkan, difokuskan dengan lensa, atau dipantulkan dari permukaan dari permukaan yang dapat memantulkan. Spektrum dari radiasi ini antara 0,7 s.d. 1000  $\mu\text{m}$ . Karenanya, radiasi ini tidak dapat dilihat dengan mata telanjang. Area ini berada pada frekuensi yang lebih kecil dari cahaya tampak sehingga disebut dengan inframerah. Ilustrasi pancaran inframerah dari suatu objek ditunjukkan pada Gambar 2.2.

Sebagian besar rancangan dasar dari sebuah termometer inframerah terdiri dari sebuah lensa untuk memfokuskan energi inframerah pada suatu detektor, yang mengubah energi menjadi sebuah sinyal listrik [12]. Konfigurasi ini memudahkan pengukuran suhu dari suatu jarak tertentu tanpa kontak dengan objek yang diukur. Sebagai contoh, termometer inframerah berguna untuk mengukur suhu dalam keadaan di mana termokopel atau sensor bertipe *probe* lain tidak dapat digunakan atau tidak dapat menghasilkan data yang akurat karena beberapa sebab.

Salah satu parameter dari pengukuran adalah *field of view* (FOV). Semakin kecil FOV, maka semakin kecil juga area (luas) pengukuran.

Sehingga semakin kecil FOV maka semakin akurat pengukuran yang dilakukan. Untuk mengubah besarnya FOV dapat digunakan lensa Fresnel. Dalam *paper* ini [11], disebutkan bahwa FOV dapat dirumuskan sebagai berikut:

$$FOV_{half-angle} \approx \left| \tan^{-1} \left( \frac{d_s/2}{f} \right) \right| \quad (2.1)$$

Di mana  $d_s$  adalah diameter area penyerapan (*absorber area*) dari sensor thermopile dan  $f$  adalah parameter *focal length* dari lensa Fresnel.

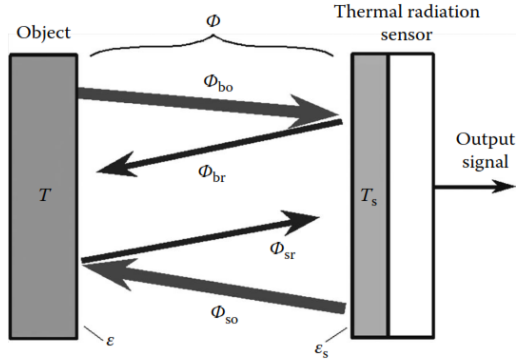
### 2.3. Sensor Radiasi Inframerah

Salah satu kelas dari sensor (detektor) radiasi inframerah disebut detektor termal. Berlawanan dengan detektor kuantum yang merespons foton individual, detektor termal merespons panas yang diakibatkan oleh penyerapan radiasi termal oleh permukaan elemen penginderaan yang mampu mengubah tingkat panas menjadi sinyal listrik [14]. Panas yang diserap meningkatkan suhu permukaan, dan kenaikan suhu yang dihasilkan menjadi ukuran radiasi termal yang terjadi. Detektor termal jauh lebih sensitif daripada detektor kuantum karena untuk menaikkan suhu detektor ke tingkat yang cukup, dibutuhkan banyak foton, sementara detektor kuantum menghasilkan sebuah elektron (dan kadang-kadang banyak) untuk foton tunggal yang terserap. Tidak seperti detektor kuantum, respons detektor termal rata di atas semua rentang spektral.

Berikut adalah perilaku detektor radiasi termal saat terkena benda. Gambar 2.3. adalah ilustrasi suatu objek yang memiliki suhu  $T$  dan emisivitas  $\varepsilon$ . Objek tersebut menghadap sebuah detektor yang memiliki suhu  $T_s$  dan emisivitas  $\varepsilon_s$ . Perhatikan bahwa objek memancarkan fluks termal  $\Phi_{bo}$  ke arah sensor sementara sensor memancarkan fluksnya sendiri ke arah objek. Aliran radiasi bidireksional ini melewati saluran kopling yang memiliki luas penampang melintang  $A$ . Menurut Hukum Stefan-Boltzmann, fluks ini dapat dinyatakan sebagai:

$$\begin{aligned} \Phi_{bo} &= A\varepsilon\sigma T^4 \\ \Phi_{so} &= -A\varepsilon_s\sigma T_s^4 \end{aligned} \quad (2.2)$$

Tanda negatif menunjukkan arah fluks yang berlawanan dari detektor  $\Phi_{bo}$  berkenaan dengan fluks  $\Phi_{bo}$  dari objek.



**Gambar 2.3.** Pertukaran Radiasi Panas [14]

Bagian yang lebih besar dari fluks  $\Phi_{bo}$  diserap oleh detektor yang memiliki absorptivitas (emisivitas)  $\varepsilon_s$ :

$$\Phi_{bs} = A\varepsilon\varepsilon_s\sigma T^4 \quad (2.3)$$

Sementara detektor menyerap energi dari objek, ia juga kehilangan energi terhadap objek tersebut. Dengan demikian, pada gilirannya, benda menyerap fluks detektor yang merambat ke arah yang berlawanan:

$$\Phi_{sb} = -A\varepsilon\varepsilon_s\sigma T_s^4 \quad (2.4)$$

Kedua fluks ini bergabung ke dalam fluks bersih akhir yang diserap atau dilepaskan oleh permukaan detektor:

$$\Phi = \Phi_{bs} + \Phi_{sb} = A\varepsilon\varepsilon_s\sigma(T^4 - T_s^4) \quad (2.5)$$

Persamaan ini membentuk hubungan antara daya termal,  $\Phi$ , diserap atau dilepaskan oleh sensor, dan suhu absolut dari objek dan sensor,  $T$  dan  $T_s$ . Fluks bersih ini menyebabkan suhu elemen penginderaan di detektor bergeser sesuai dengan itu. Bergantung pada yang lebih besar,  $T$  atau  $T_s$ , suhu elemen penginderaan akan naik atau turun. Perubahan suhu dapat digunakan untuk menentukan fluks termal bersih. Untuk mengukur pergeseran suhu detektor, elemen penginderaan memiliki sensor suhu tertanam yang menghasilkan sinyal listrik keluaran yang sebanding dengan fluks bersih,  $\Phi$ , yang dijelaskan oleh Persamaan (2.5).

Sinyal keluaran detektor berupa tegangan dapat didefinisikan sebagai:

$$V = g\Phi = A\varepsilon\varepsilon_s\sigma(T^4 - T_s^4) \quad (2.6)$$

dimana  $g$  adalah faktor konversi fluks termal insiden terhadap tegangan keluaran sensor. Faktor konversi umumnya adalah fungsi dari suhu sensor  $T_s$ , meskipun untuk sensor yang dirancang dengan baik, ketergantungan ini kecil.

Suhu (dalam Kelvin) objek dapat dihitung dari persamaan terbalik (2.6) [14]:

$$T_b = \sqrt[4]{T_{sc}^4 + \frac{V}{Ag\varepsilon\varepsilon_s\sigma}} \quad (2.7)$$

Di mana:

- $T_b$  : Suhu objek (K)
- $T_{sc}$  : Suhu sensor (K)
- $V$  : Tegangan output sensor (V)
- $A$  : Luas permukaan ( $m^2$ )
- $g$  : Sensivitas (V/W)
- $\varepsilon$  : Emisivitas objek
- $\varepsilon_s$  : Emisivitas sensor
- $\sigma$  : Konstanta Stefan-Boltzman ( $5,67 \times 10^{-8} \text{ W.m}^{-2}.\text{K}^{-4}$ )

Dalam rentang suhu sensor yang relatif sempit, penyebutnya (denominator) adalah konstanta yang nilainya ditemukan selama kalibrasi termometer, sedangkan  $T_{sc}$  dan  $T_b$  adalah suhu sensor referensi yang dihitung dan suhu objek. Temperatur yang dihitung ini mungkin agak berbeda dari suhu aktual  $T_s$  dan  $T$  pada Persamaan (2.6) karena ketidakpastian pengukuran yang tidak terelakkan.

Jika  $T$  dianggap dekat dengan  $T_s$ , Persamaan (2.7) dapat disederhanakan menjadi:

$$T_b \approx T_{sc} + \frac{V}{4Ag\varepsilon\varepsilon_s\sigma T_{sc}^3} \quad (2.8)$$

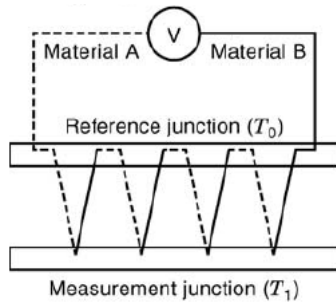
Jadi, untuk menghitung suhu objek, kita perlu mengetahui dua variabel, suhu detektor dan tegangan keluaran detektor. Faktor kalibrasi (penyebut) ditemukan selama kalibrasi termometer.

### 2.3.1. Thermopile

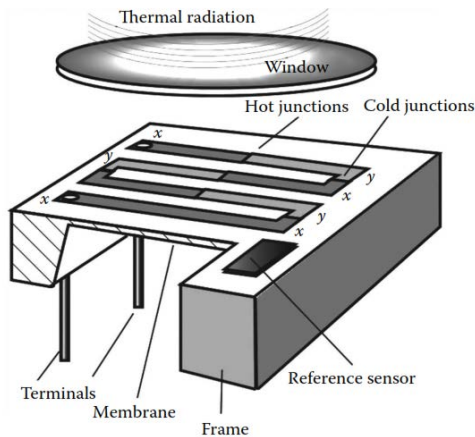
Ada sejumlah detektor termal yang dapat digunakan untuk mengukur perubahan suhu sensor yang dijelaskan. Detektor seperti itu tertanam ke permukaan sensor inframerah. Salah satu contohnya adalah sensor tipe thermistor yang dikenal sebagai bolometer. Tapi sejauh ini, yang paling populer adalah yang disebut thermopile. Thermopile termasuk dalam kelas detektor inframerah pasif, yaitu menghasilkan output listrik sehubungan dengan fluks termal yang diserap tanpa memerlukan sumber energi tambahan (tidak seperti bolometer). Prinsip operasinya sama dengan termokopel.

Sensor inframerah thermopile terdiri dari thermocouples yang terhubung secara seri dan sebuah penyerap inframerah. Karena lapisan penyerapan menyerap daya inframerah yang dipancarkan dari target yang panas, perbedaan suhu antara *hot junctions* dan *cold junctions* dari thermocouples terbentuk dan kemudian tegangan termoelektrik antara dua terminal dari termokopel yang dihubungkan seri terbentuk seperti yang ditunjukkan pada Gambar 2.4. Tegangan keluaran dari thermopile secara langsung proporsional terhadap daya inframerah yang terserap, di mana adalah sebuah fungsi suhu dari pemancar inframerah dan suhu lingkungan dari thermopile [15].

Awalnya, diciptakan oleh Joule untuk meningkatkan sinyal output dari sensor termoelektrik. Dia menghubungkan beberapa termokopel dalam rangkaian dan semua *hot junction* tergabung bersama secara termal dan *cold junction* terpisah semua. Saat ini, thermopile memiliki konfigurasi yang berbeda. Aplikasi utamanya adalah deteksi radiasi termal. Detektor thermopile menghasilkan sinyal yang hampir proporsional dengan gradien suhu antara *hot junction* dan *cold junction* namun, tidak seperti bolometer, tidak responsif terhadap suhu absolut. Itulah sebabnya thermopile harus ditambahkan oleh sensor suhu absolut, misalnya thermistor atau sensor suhu semikonduktif [16]. Sensor absolut tambahan ini kadang disebut sensor suhu referensi. Hal ini penting untuk memantau suhu  $T_s$  jika Persamaan (2.7) atau (2.8) akan digunakan untuk menghitung suhu  $T_b$  dari suatu objek.



**Gambar 2.4.** Diagram Skematik Thermopile [15]



**Gambar 2.5.** Penampang Melintang Sensor Thermopile [14]

Tampilan dari sensor inframerah thermopile ditunjukkan pada Gambar 2.5. Sensor terdiri dari *frame* yang memiliki massa termal yang relatif besar. Biasanya dibuat dari silikon. Untuk lebih menstabilkannya secara termal, *frame* biasanya terikat pada *metal header*. *Frame* adalah tempat di mana *cold junction* dan sensor suhu referensi diletakkan. *Frame* tersebut mendukung membran tipis (pada orde 1  $\mu\text{m}$ ) yang kapasitas termalnya dan konduktivitas termalnya kecil, sehingga suhunya bisa berubah dengan cepat dan dengan jumlah yang cukup. Membran adalah area dimana semua *hot junction* diposisikan. Membran menyerap radiasi termal insiden, sehingga suhu bergeser dari  $T_s$  dan selanjutnya terdeteksi



oleh *junction thermopile*. Meskipun thermopile hanyalah satu bagian dari sensor inframerah termal ini, keseluruhan sensor disebut "thermopile".

## 2.4. Thermistor

Istilah thermistor adalah gabungan dari kata termal dan resistor. Nama ini biasanya diaplikasikan pada sensor oksida logam yang dibuat dalam bentuk *droplets*, batangan, silinder, serpihan persegi, dan film tebal. Termistor termasuk dalam kelas sensor suhu absolut. Artinya, ia dapat mengukur suhu yang direferensikan pada skala suhu mutlak. Semua termistor dibagi menjadi dua kelompok: NTC (*Negative Temperature Coefficient*) dan PTC (*Positive Temperature Coefficient*). Hanya thermistor NTC yang berguna untuk pengukuran suhu presisi [17]. Thermistor oksida logam konvensional memiliki NTC. Artinya, resistansinya menurun seiring dengan kenaikan suhu. Resistansi termistor NTC, seperti resistor apapun, ditentukan oleh dimensi fisik dan resistivitas material.

Untuk menggunakan termistor pada perangkat yang sebenarnya, fungsi transfernya (ketergantungan suhu dari suatu resistansi) harus ditetapkan secara akurat. Karena fungsi ini sangat non-linier dan umumnya spesifik untuk setiap perangkat tertentu, persamaan analitis yang menghubungkan resistansi dan suhu sangat diinginkan. Beberapa model matematis dari fungsi transfer termistor telah diusulkan. Namun harus diingat, bagaimanapun, bahwa model apapun hanyalah perkiraan dan, umumnya, semakin sederhana modelnya, semakin rendah keakuratannya. Di sisi lain, untuk model yang lebih kompleks, kalibrasi dan penggunaan termistor menjadi lebih sulit. Semua model sekarang didasarkan pada fakta eksperimental yang membuktikan bahwa logaritma resistansi thermistor  $S$  berhubungan dengan suhu absolutnya  $T$  dengan persamaan polinomial:

$$\ln S = A_0 + \frac{A_1}{T} + \frac{A_2}{T^2} + \frac{A_3}{T^3} \quad (2.9)$$

Dari persamaan dasar ini, tiga model komputasi telah diusulkan Sederhana, Fraden, dan Steinhart dan Hart. Steinhart-Hart pada tahun 1968 mengajukan model untuk rentang oseanografi dari  $-3^\circ\text{C}$  sampai  $30^\circ\text{C}$  yang sebenarnya berguna untuk rentang yang jauh lebih luas. Model ini didasarkan pada Persamaan (2.9), dari mana suhu dapat dihitung sebagai

**Tabel 2.2.** Penggunaan Praktis Thermistor Model Steinhart-Hart [17]

Parameter	Model Steinhart-Hart
Kesalahan Maksimal pada 0 s.d. 70 °C	$\pm 0,003 \text{ }^{\circ}\text{C}$
Jumlah Kalibrasi	3
Fungsi Resistansi (S) Terhadap Suhu	$S = e^{\left(A_0 + \frac{A_1}{T} + \frac{A_2}{T^2} + \frac{A_3}{T^3}\right)}$ (2.12)
Titik Kalibrasi	$S_1 \text{ pada } T_1, S_2 \text{ pada } T_2, S_3 \text{ pada } T_3$
Komputasi Analisis Fungsi Suhu (dalam Kelvin) Terhadap 3 Titik Kalibrasi ( $S_n$ dan $T_n$ )	$T = [A + B(\ln S) + C(\ln S)^3]^{-1}$ (2.13)
	$L_1 = \ln(S_1) \quad Y_1 = \frac{1}{T_1}$
	$L_2 = \ln(S_2) \quad Y_2 = \frac{1}{T_2}$
	$L_3 = \ln(S_3) \quad Y_3 = \frac{1}{T_3}$
	$\gamma_2 = \frac{Y_2 - Y_1}{L_2 - L_1} \quad \gamma_3 = \frac{Y_3 - Y_1}{L_3 - L_1}$ (2.14)
	$C = \left(\frac{\gamma_3 - \gamma_2}{L_3 - L_2}\right) (L_1 + L_2 + L_3)^{-1}$
	$B = \gamma_2 - C(L_1^2 + L_1L_2 + L_2^2)$
	$A = Y_1 - (B + L_1^2C)L_1$

$$T = [\alpha_0 + \alpha_1 \ln S + \alpha_2 (\ln S)^2 + \alpha_3 (\ln S)^3]^{-1} \quad (2.10)$$

Steinhart dan Hart menunjukkan bahwa kuadrat dapat dihilangkan tanpa kehilangan akurasi; Dengan demikian, persamaan akhir menjadi persamaan (2.11). Untuk komputasi dapat digunakan persamaan (2.13) dan (2.14) yang ada pada Tabel 2.2

$$T = [b_0 + b_1 \ln S + b_3 (\ln S)^3]^{-1} \quad (2.11)$$

#### 2.4.1. Instrumentasi Thermistor

Untuk akurasi tertinggi dan untuk rentang suhu lebih dari beberapa derajat, resistansi termistor paling baik diukur secara langsung dengan menggunakan voltmeter digital berkualitas baik yang dikombinasikan dengan persamaan thermistor. Perhatian harus diberikan untuk memperhitungkan efek resistansi *lead* dan tegangan *offset* dan untuk memastikan *self-heating* termistor karena arus penginderaan tidak berlebihan. Dengan memperhatikan desain pengukuran, pemilihan termistor, penggunaan, dan kalibrasi, akurasi mendekati 0,001 °C dapat dicapai. Thermometer berbasis termistor tersedia secara komersial dengan tingkat akurasi ini.

Untuk rentang suhu yang sempit, dan resolusi tertinggi, rangkaian pengukuran yang dibuat dengan tujuan dapat menguntungkan. Tegangan keluaran rangkaian pada Gambar 2.6 adalah [18]:

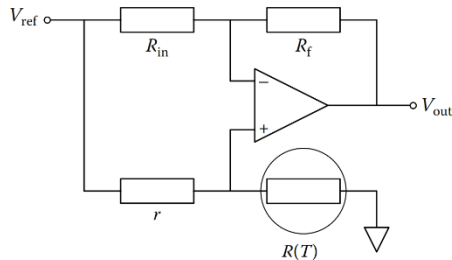
$$V_{out} = \frac{R(T)R_{in} - rR_f}{R_{in}[r + R(T)]} V_{ref} \quad (2.15)$$

dimana resistor seperti ditunjukkan. Kondisi untuk tegangan output nol adalah:

$$\frac{R_{in}}{R_f} = \frac{r}{R(T_0)} \quad (2.16)$$

dimana  $T_0$  adalah suhu operasi nominal dari termometer. Tegangan output adalah fungsi nonlinear dari resistansi termistor, dan nonlinearitas ini digunakan untuk mengkompensasi nonlinearitas termistor. Ditemukan bahwa nonlinearitas sebagian besar dihilangkan jika  $r$  dipilih sebagai berikut:

$$r = \frac{\beta T_0 - 2T_0}{\beta T_0 + 2T_0} R(T_0) \quad (2.17)$$



**Gambar 2.6.** Rangkaian Instrumentasi Thermistor [18]

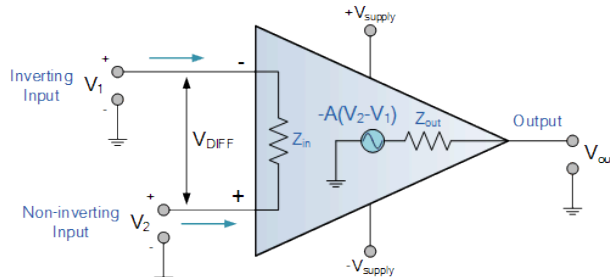
## 2.5. Rangkaian Amplifier

### 2.5.1. Dasar Operational Amplifier

*Operational Amplifier* (Op-Amp) adalah perangkat linier yang memiliki semua sifat yang dibutuhkan untuk penguatan DC yang hampir ideal dan karena itu digunakan secara ekstensif dalam pengondisian sinyal, penyaringan atau untuk melakukan operasi matematis seperti penambahan, pengurangan, integrasi, dan diferensiasi.

Op-Amp pada dasarnya adalah perangkat penguat tegangan yang dirancang untuk digunakan dengan komponen umpan balik eksternal seperti resistor dan kapasitor antara terminal output dan inputnya. Komponen umpan balik ini menentukan fungsi atau penguat yang dihasilkan dari penguat dan berdasarkan konfigurasi umpan balik yang berbeda apakah resistif, kapasitif, atau keduanya, penguat dapat melakukan berbagai operasi yang berbeda, sehingga menghasilkan namanya sebagai "*Operational Amplifier*".

Op-Amp pada dasarnya adalah perangkat tiga terminal yang terdiri dari dua masukan impedansi tinggi, yang disebut Masukan Pembalik, ditandai dengan tanda negatif atau "minus", (-) dan yang lainnya disebut Masukan Non-Pembalik, ditandai dengan tanda positif atau "plus" (+). Terminal ketiga mewakili port Keluaran Op-Amp yang dapat menyerap dan menyuplai berupa tegangan atau arus. Dalam Op-Amp linear, sinyal output adalah faktor amplifikasi, yang dikenal sebagai penguatan amplifier ( $A$ ) dikalikan dengan nilai sinyal input dan tergantung pada sifat sinyal input dan output.



**Gambar 2.7.** Rangkaian Ekuivalen Op-Amp Ideal [21]

Berikut adalah parameter dari Op-Amp ideal seperti pada Gambar 2.7:

- *Open Loop Gain*,  $(A_{VO}) = \infty$   
Fungsi utama penguat operasional adalah untuk memperkuat sinyal input dan loop gain yang lebih besar yang dimiliki maka lebih baik. Loop gain terbuka adalah penguatan dari op-amp tanpa umpan balik positif atau negatif dan untuk penguat semacam itu gain akan tak terbatas namun nilai sebenarnya berkisar antara 20.000 s.d. 200.000.
- *Input impedance*,  $(Z_{IN}) = \infty$   
Impedansi masukan adalah perbandingan tegangan masukan terhadap arus masukan dan diasumsikan tak terbatas untuk mencegah arus yang mengalir dari sumber daya ke dalam rangkaian masukan penguat ( $I_{IN} = 0$ ). Op-amp sebenarnya memiliki arus bocor masukan dari beberapa pico-ampere s.d. beberapa mili-ampere.
- *Output impedance*,  $(Z_{OUT}) = 0$   
Impedansi output dari Op-amp ideal diasumsikan nol yang berfungsi sebagai sumber tegangan internal yang sempurna tanpa resistansi internal sehingga dapat memasok arus sebanyak yang diperlukan pada beban. Resistansi internal ini efektif secara seri dengan beban sehingga mengurangi tegangan output yang tersedia pada beban. Op-amp sebenarnya memiliki impedansi keluaran dalam kisaran 100-20 k $\Omega$ .
- *Bandwidth*,  $(BW) = \infty$   
Penguat operasional yang ideal memiliki respons frekuensi yang tak terbatas dan dapat memperkuat sinyal frekuensi dari DC ke frekuensi AC tertinggi sehingga oleh karena itu diasumsikan memiliki bandwidth tak terbatas. Dengan op-amp sebenarnya, bandwidth

dibatasi oleh produk Gain-Bandwidth (GB), yang sama dengan frekuensi dimana penguatan amplifier sebesar satu.

- *Offset Voltage*, ( $V_{IO}$ ) = 0

Output amplifier akan menjadi nol bila selisih tegangan antara input pembalik dan non-pembalik adalah nol, sama atau bila kedua input ground. Op-amp sebenarnya memiliki sejumlah output tegangan *offset*.

Dari karakteristik "ideal" di atas, terlihat bahwa resistansi masukan tidak terbatas, jadi tidak ada arus yang masuk ke terminal input ("aturan arus") dan bahwa tegangan offset masukan diferensial adalah nol ("aturan tegangan"). Penting untuk mengingat dua sifat ini karena akan membantu dalam memahami cara kerja Op-amp berkaitan dengan analisis dan perancangan rangkaian op-amp.

Namun, Op-amp sebenarnya seperti uA741 yang umum tersedia, misalnya tidak memiliki penguatan atau *bandwidth* yang tidak terbatas namun memiliki "*Loop Gain*" yang khas yang didefinisikan sebagai o penguatan output amplifier tanpa sinyal umpan balik eksternal yang terhubung dengannya dan untuk Op-amp tipikal sekitar 100 dB di DC (nol Hz). Penguatan output ini turun secara linier dengan frekuensi turun ke "*Unity Gain*" atau 1, sekitar 1 MHz.

Penguatan tegangan ( $A_V$ ) Op-amp dapat ditemukan dengan menggunakan rumus berikut:

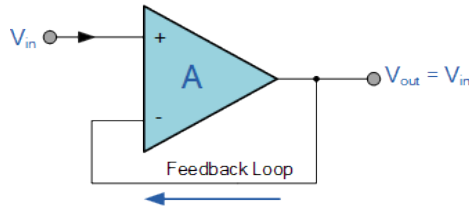
$$\text{Voltage Gain } (A_V) = \frac{V_{OUT}}{V_{IN}} \quad (2.18)$$

dan dalam Decibels (dB) diberikan sebagai:

$$\text{Voltage Gain } (A_V) = 20 \cdot \log \left( \frac{V_{OUT}}{V_{IN}} \right) \text{ dB} \quad (2.19)$$

### 2.5.2. *Non-Inverting Voltage Follower*

Dalam konfigurasi rangkaian non-pembalik ini seperti pada Gambar 2.8, impedansi masukan  $R_{in}$  telah meningkat sampai tak terhingga dan impedansi umpan balik  $R_f$  menurun menjadi nol. Keluaran dihubungkan langsung kembali ke input pembalik negatif sehingga umpan baliknya 100% dan  $V_{in}$  sama persis dengan  $V_{out}$  sehingga menghasilkan penguatan 1 atau *unity*. Sebagai tegangan masukan  $V_{in}$  diterapkan pada input non-pembalik, penguatan penguat diberikan sebagai berikut:



**Gambar 2.8.** Konfigurasi *Non-Inverting Voltage Follower* [22]

$$V_{OUT} = A \cdot (V_{IN})$$

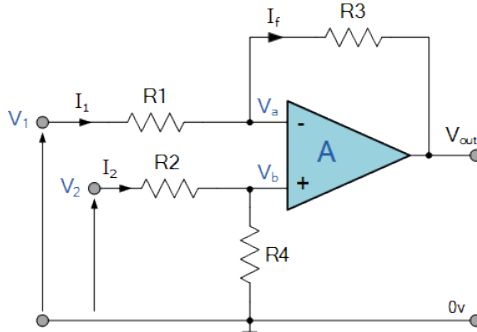
$$V_{IN} = V + \text{ dan } V_{OUT} = V -$$

Sehingga penguatan 
$$A_V = \frac{V_{OUT}}{V_{IN}} = +1 \quad (2.20)$$

Karena tidak ada arus yang mengalir ke terminal input non-pemalik, impedansi masukan tidak terbatas (op-amp ideal) dan juga tidak ada arus yang mengalir melalui *loop* umpan-balik sehingga nilai resistansi berapapun dapat ditempatkan di *loop* umpan balik tanpa mempengaruhi karakteristik rangkaian karena tidak ada arus listrik yang hilang di dalamnya, tidak ada arus mengalir, *drop* tegangan yang nol, kehilangan daya yang nol. Karena arus masukan nol memberi daya masukan nol, *voltage follower* dapat memberikan penguatan daya yang besar. Namun dalam rangkaian penyangga penguatan yang sebenarnya, sebuah resistor bernilai rendah (biasanya 1 kΩ) diperlukan untuk mengurangi arus bocor masukan offset, dan juga jika op-amp bertipe umpan balik arus. *Voltage follower* atau penyangga *unity gain* adalah jenis penguat non-pemalik yang sangat umum dan sangat berguna yang biasa digunakan dalam rangkaian elektronika sampai dengan sirkuit terisolasi satu dengan lain terutama pada variabel tingkat orde tinggi untuk memisahkan dari satu tahap saringan ke tahap yang lain.

### 2.5.3. Penguat Diferensial

Penguat diferensial memperkuat selisih antara dua tegangan yang membuat rangkaian op-amp jenis pengurang ini tidak seperti penguat penjumlah yang menambahkan atau menjumlah tegangan masukan. Jenis rangkaian op-amp ini umumnya dikenal sebagai konfigurasi penguat diferensial dan ditunjukkan Gambar 2.9.



**Gambar 2.9.** Konfigurasi Rangkaian Penguat Diferensial [23]

Dengan menghubungkan setiap masukan ke *ground* 0 V kita bisa menggunakan superposisi untuk menyelesaikan tegangan keluaran  $V_{out}$ . Kemudian fungsi alih untuk rangkaian penguat diferensial diberikan sebagai berikut:

$$I_1 = \frac{V_1 - V_a}{R_1}$$

$$I_2 = \frac{V_2 - V_b}{R_2}$$

$$I_f = \frac{V_a - (V_{out})}{R_3}$$

Titik *summing*  $V_a = V_b$

$$\text{Dan } V_b = V_2 \left( \frac{R_4}{R_2 + R_4} \right)$$

$$\text{Jika } V_2 = 0 \text{ sehingga: } V_{out(a)} = -V_1 \left( \frac{R_3}{R_1} \right)$$

$$\text{Jika } V_1 = 0 \text{ sehingga: } V_{out(b)} = V_2 \left( \frac{R_4}{R_2 + R_4} \right) \left( \frac{R_1 + R_3}{R_1} \right)$$

$$V_{out(a)} = V_{out(a)} + V_{out(b)}$$



$$V_{out} = -V_1 \left( \frac{R_3}{R_1} \right) + V_2 \left( \frac{R_4}{R_2 + R_4} \right) \left( \frac{R_1 + R_3}{R_1} \right) \quad (2.21)$$

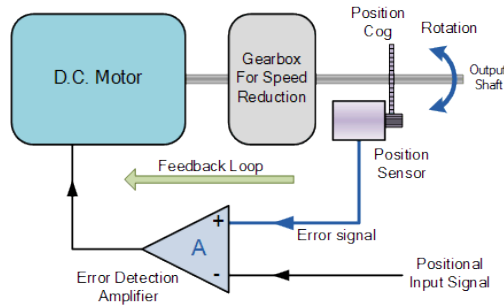
Jika resistor,  $R_1 = R_2$  dan  $R_3 = R_4$ , fungsi alih di atas untuk penguat diferensial dapat disederhanakan menjadi fungsi sebagai berikut:

$$V_{OUT} = \frac{R_3}{R_1} (V_2 - V_1) \quad (2.22)$$

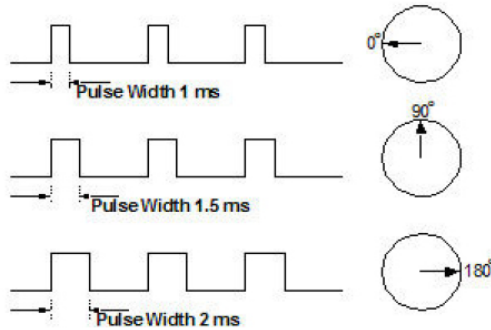
Jika semua resistor mempunyai nilai resistansi yang sama besar, di mana  $R_1 = R_2 = R_3 = R_4$ , maka rangkaian akan menjadi sebuah penguat diferensial dengan penguatan *Unity* (1) dan penguatan tegangan dari penguat akan tepat satu atau *unity*. Sehingga persamaan keluaran akan menjadi sederhana  $V_{OUT} = (V_2 - V_1)$ . Dan juga jika masukan  $V_1$  lebih besar dari input  $V_2$ , tegangan keluaran akan menjadi negatif, dan jika masukan  $V_2$  lebih besar dari input  $V_1$ , tegangan keluaran akan menjadi positif.

## 2.6. Motor Servo

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo seperti yang ditunjukkan oleh Gambar 2.10. Motor servo disusun dari sebuah motor DC, *gearbox*, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo.



**Gambar 2.10.** Blok Diagram Rangkaian Kontrol Motor Servo [24]

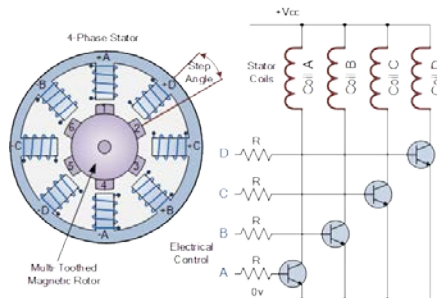


**Gambar 2.11.** Sinyal PWM dan Posisi Angular Servo [25]

Motor servo dikendalikan dengan sinyal PWM dan memberikan pergerakan yang lebih akurat dan lancar. Sinyal PWM ini terdapat pulsa minimal, pulsa maksimal, dan nilai repetisi. Dari Gambar 2.11, motor servo dapat berputar  $90^\circ$  untuk kedua arah sehingga pergerakan total adalah  $180^\circ$ . Motor servo membutuhkan pulsa setiap 20 ms dengan sebuah pulsa 1,5 ms akan membuat motor berputar ke posisi  $90^\circ$ . Pulsa kurang dari 1,5 ms membuatnya bergerak ke posisi  $0^\circ$  dan pulsa yang lebih besar dari 1,5 ms akan memutar servo pada posisi  $180^\circ$  [19].

## 2.7. Motor Stepper

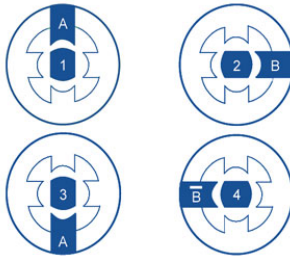
Motor stepper dianggap sebagai motor *brushless* yang dirancang secara khusus untuk masukan sinyal digital lalu menerima pulsa langsung dari mikrokontroler atau rangkaian elektronik lainnya dan berputar sesuai masukan. Motor stepper juga digunakan untuk mengendalikan putaran dengan tepat dan kecepatan dengan kendali *open-loop*. Keuntungan lain dari penggunaan motor stepper adalah kemampuannya untuk menahan beban dengan stabil ketika posisi yang diinginkan terpenuhi [20].



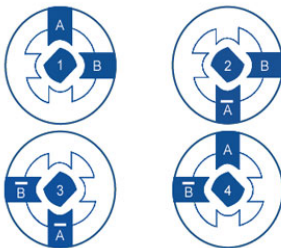
**Gambar 2.12.** Contoh Motor Stepper dengan 4 Coils Stator [24]

Contoh pada Gambar 2.12 di atas adalah motor stepper yang terdiri dari 4 *coils* medan elektromagnetik yang dilabeli oleh A, B, C, dan D. semua *coils* dengan huruf yang sama dihubungkan bersama sehingga jika diberi daya, contoh *coil* berlabel A akan menyebabkan rotor magnetik untuk meluruskan dengan pasangan *coil* tersebut. Dengan memberikan daya untuk setiap pasangan *coil* dalam urutan, rotor dapat dibuat untuk berputar atau “melangkah” (*step*) dari suatu posisi ke posisi selanjutnya dengan sebuah sudut yang ditentukan oleh konstruksi sudut step dan dengan mengaktifkan coil berurutan, rotor akan menghasilkan gerakan putar.

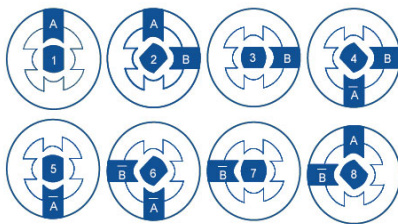
*Driver* motor stepper mengendalikan baik sudut step maupun kecepatan dari motor dengan mengaktifkan *coils* medan dalam sebuah urutan set dengan contoh, “ADCB, ADCB, ADCB, A...” dan seterusnya, rotor akan berputar dalam satu arah (*forward*) dan dengan membalikkan urutan pulsa menjadi “ABCD, ABCD, ABCD, A...” dan seterusnya, rotor akan berputar dengan arah berlawanan (*reverse*). Dalam contoh di atas, motor mempunyai 4 *coils*, membuatnya sebuah motor 4 fase, dengan jumlah kutub (*poles*) pada stator menjadi 8 ( $2 \times 4$ ) dengan jarak interval  $45^\circ$ . Jumlah dari gerigi pada rotor adalah 6 sehingga berjarak  $60^\circ$ . Kemudian terdapat kemungkinan 24 (6 gerigi  $\times$  4 *coils*) posisi atau “step” pada rotor untuk menyelesaikan satu revolusi penuh. Oleh karena itu, sudut step di atas menjadi  $360^\circ/24=15^\circ$ . Untuk mengendalikan kecepatan putaran dari motor stepper dengan mengubah waktu jeda antara pulsa digital yang diberikan pada *coils* (frekuensi), semakin lama jedaanya, semakin lambat kecepatan untuk menyelesaikan satu revolusi. Dengan memberikan sejumlah pulsa tetap kepada motor, poros motor akan berputar dengan sudut tertentu.



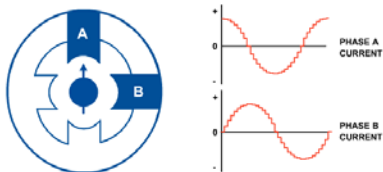
**Gambar 2.13.** Metode *Wave Drive (Full Step)* [26]



**Gambar 2.14.** Metode *2 Phases On (Full Step)* [26]



**Gambar 2.15.** Metode *Half Step* [26]



**Gambar 2.16.** Metode *Microstep* [26]

Terdapat 4 metode untuk menggerakkan motor stepper, yaitu

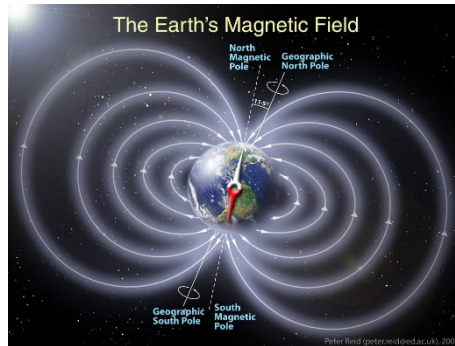
- *wave drive (full step)* disebut juga sebagai 1 *phase on*, pada metode ini hanya satu fase yang diaktifkan dalam satu waktu seperti yang ditunjukkan oleh Gambar 2.13.
- *2 phases on (full step)* selalu mengaktifkan 2 fase, sehingga rotor akan berhenti di antara 2 kutub seperti yang ditunjukkan oleh Gambar 2.14.
- *1-2 phases on (half step)* adalah gabungan dari metode *wave drive* dan *2 phase on* seperti yang ditunjukkan oleh Gambar 2.15.
- *microstep* dibuat sehingga step semakin kecil. Gelombang sinus diberikan pada kedua fase dengan beda fase antar keduanya  $90^\circ$  seperti pada Gambar 2.16.

## 2.8. Sensor Magnetometer

Magnetometer adalah alat yang mengukur medan magnet. Sebuah magnetometer adalah instrumen dengan sensor yang mengukur kerapatan fluks magnetik  $B$  (dalam satuan Tesla atau  $\text{kg}/(\text{A}\cdot\text{s}^2)$ ). Magnetometer mengacu pada sensor yang digunakan untuk mengindra medan magnet atau pada sistem yang mengukur medan magnet dengan menggunakan satu atau lebih sensor.

Karena kerapatan fluks magnetik di udara berbanding lurus dengan kekuatan medan magnet, magnetometer mampu mendeteksi fluktuasi medan bumi. Bahan yang mendistorsi garis fluks magnetik dikenal sebagai magnetik, dan termasuk bahan seperti magnetit yang memiliki medan magnet sendiri, serta konduktivitas magnetiknya sangat tinggi. Bahan semacam itu menciptakan distorsi dalam fluks magnet Bumi yang diilustrasikan pada Gambar 2.17 yang mengalir di sekitar mereka. Magnetometer mendeteksi distorsi ini.

Sebuah magnetometer mengukur kerapatan fluks magnetik pada titik di ruang di mana sensor berada. Medan magnet akan turun intensitasnya dengan jarak kubus dari objek. Oleh karena itu, jarak maksimum yang diberikan magnetometer bisa mendeteksi benda berbanding lurus dengan akar kubik sensitivitas magnetometer. Sensitivitas biasanya diukur dalam Tesla.



**Gambar 2.17.** Ilustrasi Medan Magnet Bumi [27]

Magnetometer dikelompokkan menjadi dua kategori, yaitu:

- Magnetometer vektor yang mengukur nilai kerapatan fluks pada arah tertentu dalam ruang 3 dimensi. Contohnya adalah magnetometer *fluxgate* yang dapat mengukur kekuatan setiap komponen medan bumi dengan mengarahkan sensor ke arah komponen yang diinginkan.
- Magnetometer skalar yang mengukur hanya besarnya vektor yang melewati sensor terlepas dari arahnya. Magnetometer kuantum adalah contoh dari jenis magnetometer ini.

Arah mata angin (kompas) dapat ditentukan dengan hanya menggunakan komponen  $H_x$  dan  $H_y$  dari medan magnet bumi, yaitu arah planar dengan permukaan bumi. Tahan magnetometer secara datar di area terbuka dan perhatikan pembacaan magnetik  $H_x$  dan  $H_y$ .

Nilai maksimum  $H_x$  dan  $H_y$  bergantung pada kekuatan medan magnet bumi pada saat itu. Arah mata angin (kompas) magnetik dapat ditentukan (dalam derajat) dari pembacaan sumbu x dan y magnetometer dengan menggunakan rangkaian persamaan berikut ini:

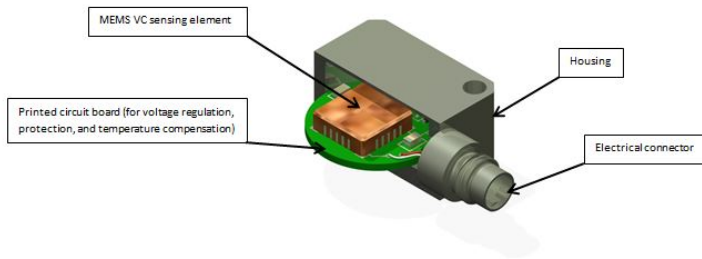
$$\begin{aligned} \text{Arah } (y > 0) &= 90 - \left[ \tan^{-1} \frac{x}{y} \right] \times \frac{180}{\pi} \\ \text{Arah } (y < 0) &= 270 - \left[ \tan^{-1} \frac{x}{y} \right] \times \frac{180}{\pi} \\ \text{Arah } (y > 0, x < 0) &= 180 \\ \text{Arah } (y < 0, x > 0) &= 0 \end{aligned}$$

## 2.9. Sensor Akselerometer

MEMS singkatan dari *micro-electro-mechanical systems* dan berlaku untuk sensor yang diproduksi dengan teknik fabrikasi mikroelektronik. Teknik ini menciptakan struktur penginderaan mekanis dengan ukuran mikroskopik, biasanya pada silikon. Bila ditambah dengan sirkuit mikroelektronika, sensor MEMS bisa digunakan untuk mengukur parameter fisik seperti percepatan. Tidak seperti sensor ICP®, sensor MEMS mengukur frekuensi sampai 0 Hz (akselerasi statis atau DC). Akselerometer kapasitif variabel (VC) MEMS memiliki rentang yang lebih rendah, perangkat bersensitivitas tinggi yang digunakan untuk pemantauan struktural dan pengukuran akselerasi konstan.

Elemen penginderaan pada akselerometer MEMS VC terdiri dari tahanan massa mikro-mesin yang tersuspensi di antara dua pelat paralel. Massa digantung pada lekukan yang menempel pada bingkai cincin. Konfigurasi ini membentuk dua kapasitor celah udara antara tahanan massa dan pelat atas dan bawah. Ketika tahanan massa bergerak saat akselerasi diterapkan, satu celah udara berkurang dan celah lainnya meningkat sehingga membuat perubahan kapasitansi sebanding dengan percepatan. Pemilihan rentang pengukuran skala penuh dicapai dengan memodifikasi kekakuan sistem suspensi dari tahanan massa. Frekuensi alami tinggi dicapai melalui kombinasi tahanan massa ringan dan kekakuan suspensi. Ketangguhan ditingkatkan melalui penggunaan penghenti mekanis pada dua lapisan luar untuk membatasi pergerakan tahanan massa.

Elemen sensor menggunakan peredam gas film terpress untuk mengurangi masukan resonansi frekuensi tinggi yang menyebabkan kejenuhan mekanis. Hal ini terjadi ketika pergerakan tahanan massa melebihi batas perpindahannya. Peredam membantu mencegah kejenuhan dengan mengurangi amplifikasi resonansi dan memperpanjang bagian datar respons frekuensi. Peredaman gas sedikit dipengaruhi oleh perubahan suhu. Elemen penginderaan dihubungkan sebagai rangkaian jembatan ke bagian elektronik lainnya di akselerometer. Ini meminimalkan kesalahan mode umum dan meningkatkan linearitas. IC ini juga mengkompensasi bias nol dan kesalahan sensitivitas terhadap suhu. Gambar 2.18 menunjukkan konstruksi akselerometer MEMS VC. Elemen penginderaan biasanya dipasang pada papan rangkaian yang ditempatkan di wadah titanium atau aluminium.



**Gambar 2.18.** Konstruksi Akselerometer VC [28]

Sudut kemiringan sistem (*tilt*) sebesar  $\rho$  dapat diukur dengan akselerometer. Vektor gaya gravitasi yang diukur oleh akselerometer dan orientasi awal dengan medan gravitasi mengarah ke bawah sesuai dengan sumbu z. jika pembacaan akselerometer adalah  $G_P$ , maka dengan tidak adanya percepatan linier:

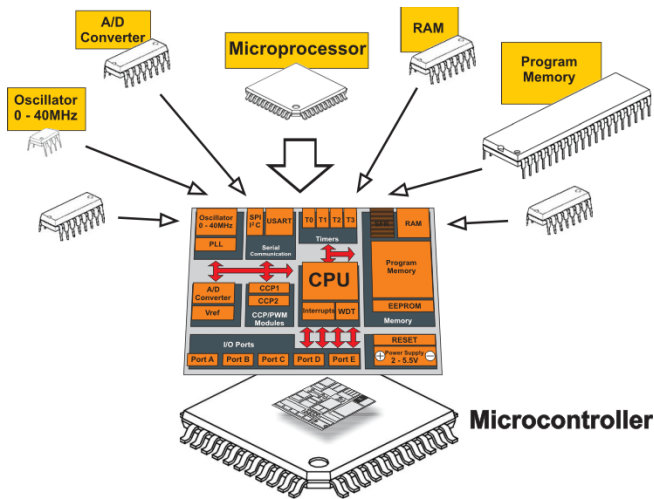
$$G_P \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = G_{PZ} = |G_P| \cos \rho \rightarrow \cos \rho = \frac{G_{PZ}}{\sqrt{G_{PX}^2 + G_{PY}^2 + G_{PZ}^2}}$$

## 2.10. Mikrokontroler

Mikrokontroler adalah perangkat elektronik milik keluarga mikrokomputer. Mikrokontroler ini dibuat dengan menggunakan teknologi VLSI pada satu *chip*. Mikrokontroler yang tersedia di pasar sekarang dengan *word-length* yang berbeda, mulai dari 4 bit, 8 bit, 64 bit sampai dengan 128 bit.

Dalam pengertian yang lebih luas, komponen yang merupakan mikrokontroler adalah memori, periferal dan yang paling penting prosesor. Mikrokontroler hadir pada perangkat dimana pengguna harus menggunakan kontrol tingkat tinggi. Mereka dirancang dan diimplementasikan untuk menjalankan fungsi tertentu seperti menampilkan bilangan bulat atau karakter pada modul layar LCD dalam alat rumah. Penggunaan mikrokontroler sangat beragam. Dalam istilah yang lebih sederhana, gawai atau peralatan apa pun yang harus berurusan dengan fungsi seperti mengukur, mengendalikan, menampilkan dan menghitung nilainya terdiri dari *chip* mikrokontroler di dalamnya. Arsitektur mikrokontroler ditunjukkan pada Gambar 2.19.





**Gambar 2.19.** Arsitektur Mikrokontroler [29]

Bagian terpenting dari sebuah mikrokontroler adalah unit pemrosesan sentral dengan *word-length* mulai dari 4-bit sampai 64-bit dan bahkan beberapa mikrokontroler modern, *word-length* melampaui batas 64-bit. *Timer* adalah satu konstituen lain dari mikrokontroler. Ada *timer watch-dog*. Ruang memori seperti RAM, ROM, EEPROM, EPROM ada untuk menyimpan data dan program. Untuk penyimpanan data, RAM volatile digunakan sedangkan untuk penyimpanan program dan operasi penyimpanan ROM dan ruang memori lainnya digunakan.

- CPU

Dianggap sebagai otak mikrokontroler, unit pemrosesan sentral menerima, menerjemahkan, dan mengeksekusi instruksi. CPU mengkoordinasikan berbagai aktivitas yang terjadi di mikrokontroler.

- Port I/O

Ada beberapa port input / output paralel pada mikrokontroler. Mereka digunakan untuk antarmuka berbagai periferal seperti *printer*, memori eksternal, LED, dan LCD ke mikrokontroler. Selain port paralel, ada port serial untuk menghubungkan periferal terhubung secara serial dengan mikrokontroler.

- Memori

Seperti halnya mikroprosesor, mikrokontroler memiliki ruang untuk memori seperti RAM, ROM termasuk EEROM dan EPROM. Ini juga

mengalokasikan sejumlah memori *flash* untuk menyimpan kode sumber program.

- *Timers and Counters*  
Ini adalah bagian penyusun mikrokontroler yang paling menarik. *Timer* dan *counter* berguna dalam operasi yang meliputi modulasi, fungsi jam, pembangkitan frekuensi, dan pengukuran, dan pembangkitan pulsa.
- *Analog to Digital Converters (ADC)*  
Konverter seperti itu berguna saat mengubah keluaran sensor yang berbentuk analog.
- *Digital to Analog Converters (DAC)*  
Bekerja dari DAC hanyalah kebalikan dari konverter analog ke digital. Seperti sudah jelas, keluarannya akan menjadi sinyal analog yang bisa digunakan untuk mengendalikan periferal analog seperti motor.

Fitur dari sebuah mikrokontroler:

- Keuntungan utama arsitektur CISC (*complex instruction set computer*), dimana mikrokontroler modern dibangun, adalah instruksi tipe makro. Instruksi makro dapat digunakan dalam program yang mengganti sejumlah instruksi.
- Mikrokontroler terbaru dioperasikan pada konsumsi daya yang lebih rendah. Biasanya mereka dapat mendukung tegangan kerja 1,8-5,5 V.
- Memori tingkat lanjut adalah fitur lain dari mikrokontroler. Penggunaan memori ROM seperti EEPROM dan EPROM (*flash memory*) membuatnya lebih andal dan *user-friendly*. Sementara EEPROM adalah memori yang relatif lambat, EPROM lebih cepat. Fakta bahwa hal itu memungkinkan lebih banyak menghapus / menulis siklus juga membuatnya lebih bermanfaat.

Keuntungan utama sebuah mikrokontroler adalah bahwa biaya rendah dengan semua bagian dipasang bersamaan pada satu chip. Desainnya membuatnya lebih padat dan mudah digunakan. Kemudahan penggunaan mikrokontroler dan proses perawatan yang relatif mudah juga membuatnya lebih andal. Hampir semua pin di mikrokontroler diprogram dan itu membuat mikrokontroler sangat *user-friendly*. Kesederhanaan saat berinteraksi ROM, RAM dan port I/O. Kemudahan pemecahan masalah dan waktu persyaratan minimal untuk berbagai operasi merupakan keuntungan penting lainnya.

Karena berisi semua komponen pada satu *chip*, kekurangan mikrokontroler memiliki arsitektur yang relatif kompleks. Mikrokontroler tidak cocok untuk antarmuka perangkat daya tinggi secara langsung dan mereka hanya dapat melakukan sejumlah operasi secara bersamaan.

## 2.11. Microsoft Visual Studio

Microsoft Visual Studio adalah lingkungan pengembangan terpadu (*Integrated Development Environment / IDE*) dari Microsoft. Ini digunakan untuk mengembangkan program komputer, serta situs web, aplikasi web, layanan web, dan aplikasi *mobile*. Visual Studio menggunakan platform pengembangan perangkat lunak Microsoft seperti Windows API, Windows Forms, Windows Presentation Foundation, Windows Store dan Microsoft Silverlight. Ini bisa menghasilkan *native-code* dan *managed-code*.

Visual Studio, yang berlogo pada Gambar 2.20, menyertakan editor kode yang mendukung IntelliSense (komponen penyelesaian kode) serta *refactoring* kode. *Debugger* terintegrasi bekerja baik sebagai *debugger* tingkat sumber dan *debugger* tingkat mesin. Alat built-in lainnya termasuk kode *profiler*, perancang *form* untuk membangun aplikasi GUI, desainer web, desainer *class*, dan perancang skema *database*. Ini menerima plug-in yang meningkatkan fungsinya di hampir setiap level - termasuk menambahkan dukungan untuk sistem kontrol sumber (seperti Subversion) dan menambahkan *toolsets* baru seperti editor dan perancang visual untuk bahasa domain khusus atau perangkat untuk aspek lain dari siklus pengembangan perangkat lunak (seperti klien Team Foundation Server: Team Explorer).

Visual Studio mendukung 36 bahasa pemrograman yang berbeda dan memungkinkan editor kode dan *debugger* untuk mendukung (dalam berbagai tingkat) hampir semua bahasa pemrograman, asalkan layanan bahasa khusus ada. Bahasa *built-in* termasuk C, C ++, CLI, Visual Basic.NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, dan CSS. Dukungan untuk bahasa lain seperti Python, Ruby, Node.js, dan M antara lain tersedia melalui *plug-in*. Java (dan J #) didukung di masa lalu.

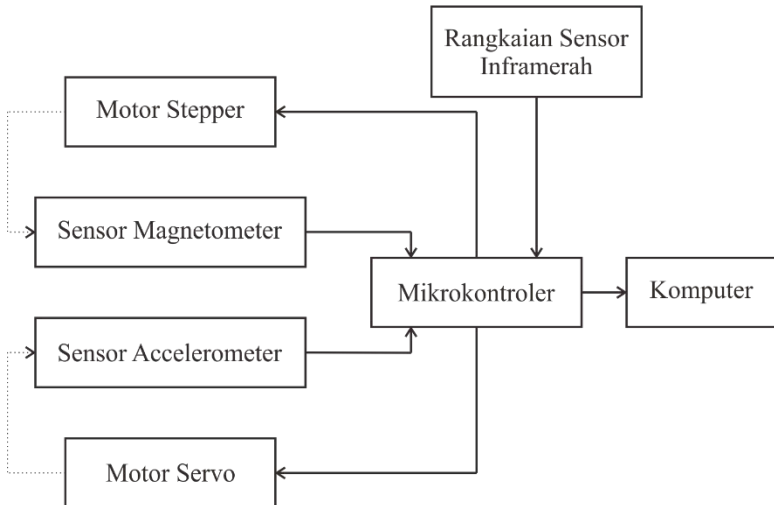
*# Halaman ini sengaja dikosongkan #*

## BAB III PERANCANGAN SISTEM

### 3.1. Blok Diagram

“Sistem Pemetaan Suhu Permukaan Lahan Menggunakan Sensor Inframerah untuk Pendeteksi Dini Kebakaran” mempunyai sistem kontrol utama mikrokontroler berupa board Arduino Nano. Mikrokontroler ini digunakan untuk mengubah nilai analog menjadi digital besar suhu yang diukur oleh rangkaian sensor inframerah dengan fasilitas ADC, mengatur putaran motor stepper dengan fungsi digital output-nya, mengatur putaran motor servo dengan fasilitas PWM-nya, membaca nilai dari sensor akselerometer dan magnetometer melalui protokol I2C, dan mengirim data suhu yang telah terukur ke komputer dengan komunikasi serial. Blok diagram sistem ini dapat dilihat pada Gambar 3.1.

Rangkaian sensor inframerah digunakan untuk menguatkan (pengondisian sinyal) tegangan keluaran dari sensor inframerah thermopile. Rangkaian ini terdiri dari penguat diferensial.



**Gambar 3.1.** Blok Diagram Sistem



### 3.2. Rangkaian Regulator Catu Daya

Rangkaian ini digunakan untuk mengatur tegangan input dari baterai menjadi 5 volt. Rangkaian ini menggunakan IC LM7805 regulator tegangan. Tegangan output 5 volt ini digunakan untuk suplai tegangan motor stepper dan motor servo yang memerlukan arus lebih besar dari pada tegangan pin +5V pada Arduino Nano. Rangkaian regulator dapat dilihat pada Gambar 3.3.

### 3.3. Rangkaian Board Mikrokontroler Arduino

Arduino Nano adalah papan pengembangan (*development board*) mikrokontroler yang berbasis chip ATmega328P dengan bentuk yang sangat mungil. Arduino Nano memiliki 14 buah digital pin yang dapat digunakan sebagai input atau output. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki resistor *pull-up* sekitar 20-50k ohm (secara *default* dalam posisi *disconnect*). Nilai maksimal adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroler. Bentuk fisik Arduino Nano ditunjukkan pada Gambar 3.4. Gambar 3.5 menunjukkan pinout dari Arduino Nano.

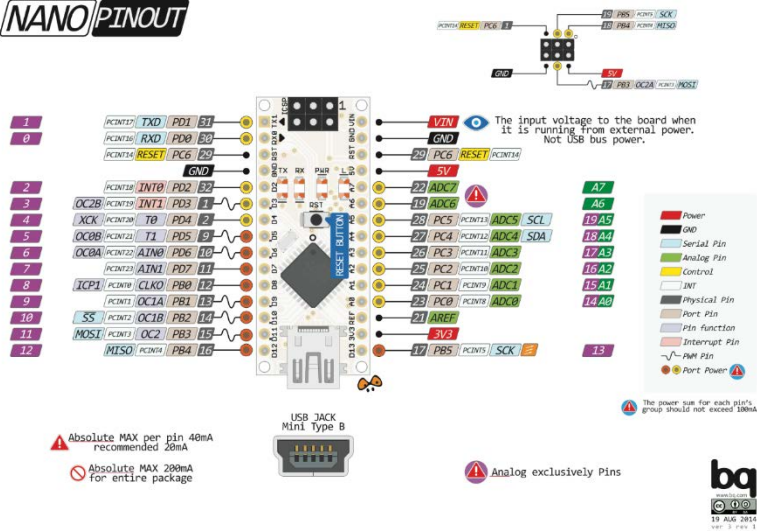
Beberapa pin memiliki fungsi khusus:

- Serial, terdiri dari 2 pin: pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
- PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan output PWM 8-bit SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan SPI Library



**Gambar 3.4.** Bentuk Fisik Arduino Nano

## NANO PINOUT

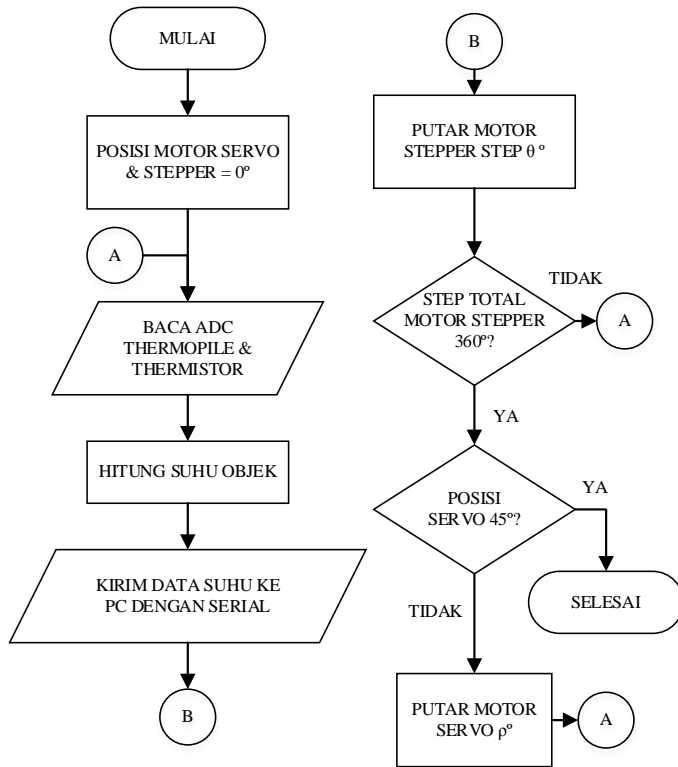


**Gambar 3.5.** Pinout dari Arduino Nano

**Tabel 3.1.** Fungsi Pin I/O pada Arduino dalam Sistem

No.	Pin Arduino	Fungsi Pin
1.	VIN	Suplai daya masukan untuk Arduino
2.	5V	Suplai daya keluaran untuk rangkaian pengondisi sinyal sensor inframerah thermopile
3.	3V3	Suplai daya keluaran untuk sensor magnetometer dan akselerometer
4.	GND	Ground
5.	GND2	Ground
6.	A2	ADC untuk tegangan keluaran rangkaian pengondisi sinyal thermistor
7.	A3	ADC untuk tegangan keluaran rangkaian pengondisi sinyal thermopile
8.	A4	SDA protokol I2C
9.	A5	SDL protokol I2C
11.	D5	Keluaran digital untuk motor stepper (jingga)
12.	D6	Keluaran digital untuk motor stepper (kuning)
13.	D7	Keluaran digital untuk motor stepper (merah muda)
14.	D8	Keluaran digital untuk motor stepper (biru)



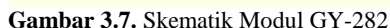


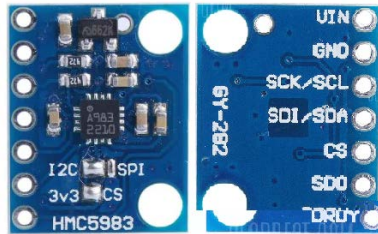
**Gambar 3.6.** Diagram Alir Program pada Arduino Nano

Dalam sistem ini beberapa pin yang digunakan fungsinya yang dijelaskan pada Tabel 3.1. Diagram alir program pada Arduino Nano ditunjukkan pada Gambar 3.6. Program dimulai dengan inisialisasi sudut pada motor servo dan stepper. Kemudian dilakukan pembacaan tegangan keluaran rangkaian pengondisi sinyal thermopile dan thermistor dengan ADC. Hasil pembacaan ADC kemudian dikomputasi sesuai rancangan perangkat lunak pengukuran suhu. Data suhu, sudut servo, dan sudut stepper kemudian dikirim ke komputer melalui komunikasi serial. Kemudian untuk mengubah arah pengukuran, putar motor stepper sebesar  $0^\circ$ . Jika step total dari motor stepper belum  $360^\circ$  akan mengulang pembacaan ADC kembali, jika sudah maka lanjut dengan putar motor

### 3.4. Rangkaian Sensor Magnetometer

HMC5983 menggunakan teknologi Anisotropic Magnetoresistive Honeywell (AMR) yang memberikan keunggulan dibanding teknologi sensor magnetik lainnya. Anisotropika Honeywell, sensor terarah yang unggul dalam linieritas, histeresis rendah, keluaran nol dan stabilitas faktor skala melebihi suhu, dan sensitivitas sumbu silang sangat rendah. Konstruksi sensor *solid-state* ini dirancang untuk mengukur arah dan besarnya medan magnet, dari mili-gauss hingga 8 gauss.





**Gambar 3.8.** Bentuk Fisik Modul GY-282

Alamat register dan alamat *device* yang digunakan pada sistem ini antara lain:

- Alamat *Device* : 0x1E
- Register Mode : 0x02
- Pengukuran Magnetometer Sumbu X [15:8] : 0x03
- Pengukuran Magnetometer Sumbu X [7:0] : 0x04
- Pengukuran Magnetometer Sumbu Z [15:8] : 0x05
- Pengukuran Magnetometer Sumbu Z [7:0] : 0x06
- Pengukuran Magnetometer Sumbu Y [15:8] : 0x07
- Pengukuran Magnetometer Sumbu Y [7:0] : 0x08

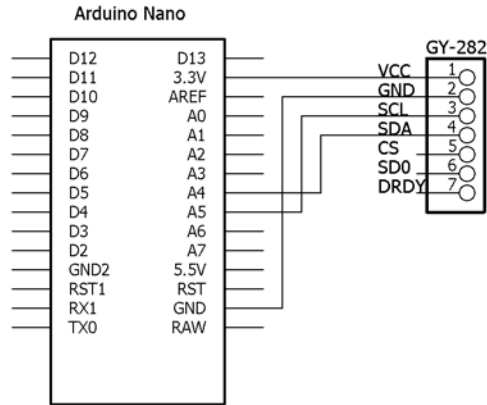
Sensor Magnetometer HMC5983 yang digunakan telah berada dalam satu modul yang dinamai GY-282 seperti pada Gambar 3.8. Skematik dari modul GY-282 ditunjukkan pada Gambar 3.7.

### 3.4.1. Perangkat Keras Sensor Magnetometer

Untuk menghubungkan modul GY-282 dengan Arduino Nano, pin VCC dihubungkan dengan pin 3V3. Untuk pin GND dihubungkan dengan pin GND. Pin SCL dihubungkan dengan pin A5, dan pin SDA dihubungkan dengan pin A4. Hubungan pin-pin antara Arduino Nano dengan modul GY-521 ditunjukkan seperti pada Gambar 3.9. dan Tabel 3.2.

**Tabel 3.2.** Hubungan Arduino Nano dengan Modul GY-282

No.	Pin GY-521	Pin Arduino	No.	Pin GY-521	Pin Arduino
1	VCC	3V3	4	SDA	A4
2	GND	GND	5	CS	-
3	SCL	A5	6	SD0	-



**Gambar 3.9.** Hubungan Arduino Nano dengan Modul GY-282

### 3.4.2. Perangkat Lunak Sensor Magnetometer

Dalam penulisan program kode untuk mendapatkan data dari sensor magnetometer, diperlukan library Wire.h untuk komunikasi antara Arduino dengan sensor magnetometer. Dari data nilai vektor medan magnet sumbu x ( $H_x$ ) dan sumbu y ( $H_y$ ), kemudian dihitung dengan rumus sebagai berikut:

$$\text{Arah } (y > 0) = 90 - \left[ \tan^{-1} \frac{x}{y} \right] \times \frac{180}{\pi}$$

$$\text{Arah } (y < 0) = 270 - \left[ \tan^{-1} \frac{x}{y} \right] \times \frac{180}{\pi}$$

$$\text{Arah } (y > 0, x < 0) = 180$$

$$\text{Arah } (y < 0, x > 0) = 0$$

Keempat kondisi di atas dalam pemrograman dapat digunakan satu fungsi saja yaitu *four-quadrant inverse tangent* atau atan2().

```
#include <Wire.h>
const char accelAddress = 0x68;
const unsigned char accelSleepMgmt = 0x6B;
const unsigned char accelXOut = 0x3B;
const char magnAddress = 0x1E;
const unsigned char magnMode = 0x02;
const unsigned char magnXOut = 0x03;
writeI2c(magnAddress, magnMode, 0x00);
data_pdu pduMagn;
readI2c(magnAddress, magnXOut, (uint8_t *)&pduMagn,
        sizeof(pduMagn));
if (i2cErr == 0){
```

```

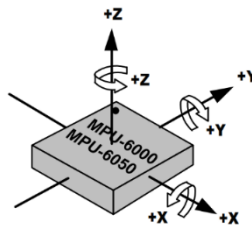
pduMagn.x = swapInt16_t(pduMagn.x);
pduMagn.y = swapInt16_t(pduMagn.y);
pduMagn.z = swapInt16_t(pduMagn.z);
float magnX = pduMagn.x * 0.00092;
float magnZ = pduMagn.y * 0.00092;
float magnY = pduMagn.z * 0.00092;
float magnHeading = atan2(magnY, magnX);
if(magnHeading < 0) magnHeading += 2*PI;
if(magnHeading > 2*PI) magnHeading -= 2*PI;
magnHeadingDegrees = magnHeading * 180/PI;
}

```

### 3.5. Rangkaian Sensor Akselerometer

MPU-60X0 adalah perangkat MotionTracking bersumbu 6 terintegrasi pertama di dunia yang menggabungkan giroskop 3 sumbu, akselerometer 3 sumbu seperti pada Gambar 3.10, dan Digital Motion Processor™ (DMP) semua dalam paket kecil 4 x 4 x 0,9 mm. Dengan bus sensor I2C khusus, langsung menerima masukan dari kompas 3 sumbu eksternal untuk menghasilkan keluaran MotionFusion™ bersumbu 9 yang lengkap. Perangkat MotionTracking MPU-60X0, dengan integrasi 6 sumbu, *on-board* MotionFusion™, dan *firmware* kalibrasi secara *run-time*. MPU-60X0 juga dirancang untuk berinteraksi dengan beberapa sensor digital non-inersial, seperti sensor tekanan, pada port I2C tambahan. MPU-60X0 kompatibel dengan keluarga MPU-30X0.

MPU-60X0 dilengkapi tiga ADC 16-bit untuk mendigitalkan keluaran giroskop dan tiga ADC 16-bit untuk mendigitalkan keluaran akselerometer. Untuk pelacakan presisi baik gerakan cepat maupun lambat, bagian-bagian ini menampilkan fitur giroskop pengguna yang dapat diprogram dengan skala penuh  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , dan  $\pm 2000$  °/detik (dps) dan akselerometer yang dapat diprogram untuk pengguna yang dapat diprogram sekitar  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , dan  $\pm 16g$ .



**Gambar 3.10.** Orientasi Sumbu Sensitivitas dan Polaritas MPU6050

Penyangga FIFO 1024 Byte *on-chip* membantu mengurangi konsumsi daya sistem dengan membiarkan prosesor sistem membaca data

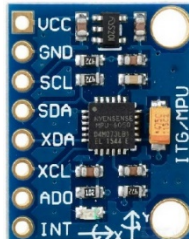
sensor secara *burst* dan kemudian masuk ke mode daya rendah saat MPU mengumpulkan data lebih banyak.

Komunikasi dengan semua register perangkat dilakukan dengan menggunakan I2C pada 400 kHz atau SPI pada 1MHz (hanya MPU-6000). Untuk aplikasi yang memerlukan komunikasi lebih cepat, register sensor dan *interrupt* dapat dibaca menggunakan SPI pada 20MHz (hanya MPU-6000). Fitur tambahan mencakup sensor suhu yang disematkan dan osilator *on-chip* dengan variasi  $\pm 1\%$  selama rentang suhu operasi.

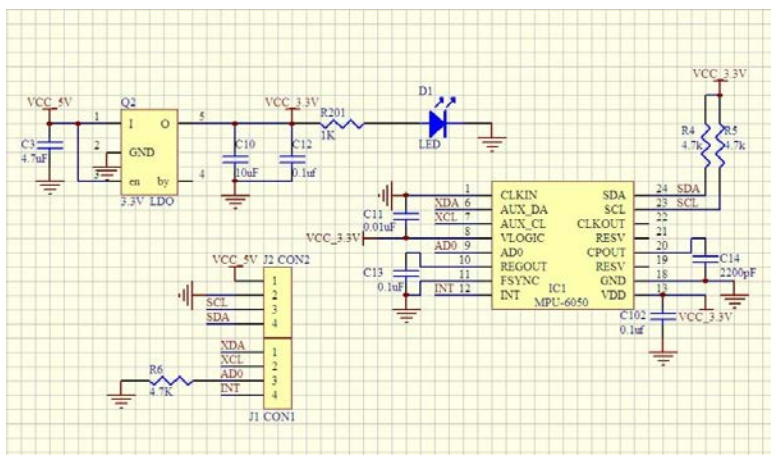
Beberapa register yang digunakan pada sistem ini antara lain:

- Alamat *Device* : 0x68
- Manajemen Daya : 0x6B
- Pengukuran Akselerometer Sumbu X [15:8]: 0x3B
- Pengukuran Akselerometer Sumbu X [7:0] : 0x3C
- Pengukuran Akselerometer Sumbu Y [15:8]: 0x3D
- Pengukuran Akselerometer Sumbu Y [7:0] : 0x3E
- Pengukuran Akselerometer Sumbu Z [15:8]: 0x3F
- Pengukuran Akselerometer Sumbu Z [7:0] : 0x40

Sensor akselerometer MPU6050 yang digunakan telah berada dalam satu modul yang dinamai GY-521. Skematik dari modul GY-521 ditunjukkan pada Gambar 3.12 sedangkan bentuk fisik dapat dilihat pada Gambar 3.11.



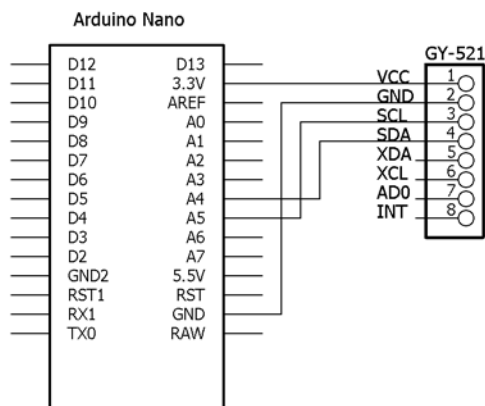
**Gambar 3.11.** Bentuk Fisik Modul GY-521



**Gambar 3.12.** Skematik Modul GY-521

### 3.5.1. Perangkat Keras Sensor Akselerometer

Untuk menghubungkan modul GY-521 dengan Arduino Nano, pin VCC dihubungkan dengan pin 3V3. Untuk pin GND dihubungkan dengan pin GND. Pin SCL dihubungkan dengan pin A5, dan pin SDA dihubungkan dengan pin A4. Hubungan pin-pin antara Arduino Nano dengan modul GY-521 ditunjukkan seperti pada Gambar 3.13. dan Tabel 3.3.



**Gambar 3.13.** Hubungan Arduino Nano dengan Modul GY-521

**Tabel 3.3.** Hubungan Arduino Nano dengan Modul GY-521

No.	Pin GY-521	Pin Arduino
1	VCC	3V3
2	GND	GND
3	SCL	A5
4	SDA	A4
5	XDA	-
6	XCL	-
7	AD0	-
8	INT	-

### 3.5.2. Perangkat Lunak Sensor Akselerometer

Dalam penulisan program kode untuk mendapatkan data dari sensor magnetometer, diperlukan *library* Wire.h untuk komunikasi antara Arduino dengan sensor akselerometer.

$$G_{PZ} = |G_P| \cos \rho$$

$$\cos \rho = \frac{G_{PZ}}{\sqrt{G_{PX}^2 + G_{PY}^2 + G_{PZ}^2}}$$

$$\rho = \cos^{-1} \left( \frac{G_{PZ}}{\sqrt{G_{PX}^2 + G_{PY}^2 + G_{PZ}^2}} \right)$$

```
#include <Wire.h>
const char accelAddress = 0x68;
const unsigned char accelSleepMgmt = 0x6B;
const unsigned char accelXOut = 0x3B;

writeI2c(accelAddress, accelSleepMgmt, 0x00);

data_pdu pduAccel;
readI2c(accelAddress, accelXOut, (uint8_t *)&pduAccel,
    sizeof(pduAccel));
pduAccel.x = swapInt16_t(pduAccel.x);
pduAccel.y = swapInt16_t(pduAccel.y);
pduAccel.z = swapInt16_t(pduAccel.z);
float accelX = pduAccel.x / 16384.0f;
float accelY = pduAccel.y / 16384.0f;
float accelZ = pduAccel.z / 16384.0f;

float tiltDen = sqrtf(accelX*accelX+accelY*accelY+accelZ*accelZ);
tiltUnfild = acos(accelY/tiltDen)*180/PI;
tiltFild = tiltCoe*tiltUnfild+(1-tiltCoe)*tiltFild;
```





**Gambar 3.14.** Bentuk Fisik Thermopile ZTP-135SR

### 3.6. Perancangan Rangkaian Sensor Inframerah

Sensor inframerah yang digunakan adalah thermopile dengan tipe ZTP-135SR. Sensor thermopile ini digunakan untuk pengukuran suhu permukaan secara non-kontak. Model ZTP-135SR terdiri dari elemen termo, filter IR datar, thermistor untuk kompensasi suhu dalam paket TO-46 (18) yang disegel secara hermet. Fitur yang ada antara lain adalah Termasuk sensor suhu lingkungan (thermistor) untuk kompensasi, sensitivitas tinggi, waktu respon cepat, dan biaya rendah. Bentuk fisik thermopile ZTP-135SR ditunjukkan pada Gambar 3.14.

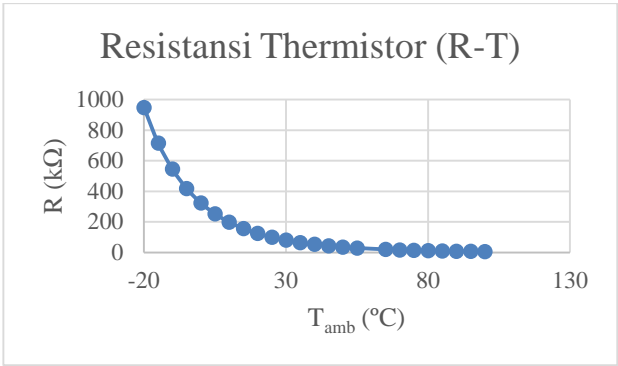
Spesifikasi thermopile ZTP-135SR adalah sebagai berikut:

- Ukuran *Chip* : 1.8 x 1.8 mm<sup>2</sup>
- Ukuran *Diaphragm* : 1.4 x 1.4 mm<sup>2</sup>
- Jumlah Kopel : 0.7 x 0.7 mm<sup>2</sup>
- Tahanan Dalam : 60 kΩ ±30% @ 77°F (25°C)
- Resistansi T.C. : <0,12 % °F (°C)
- Responsivitas : 62 ±30% V/W
- Responsivitas T.C. : -0,10 % / °F (°C)
- Tegangan *Noise* : 32 nV rms
- NEP (*Noise Equivalent Power*) : 0.51 nW/√Hz
- Detektivitas : 1.35E+08 cm√Hz/W
- Konstanta Waktu : 25 ms
- Suhu Kerja : -4°F to 212°F (-20°C to 100°C)
- Suhu Penyimpanan : -40°F to 248°F (-40°C to 120°C)
- Resistensi Thermistor : 100 kΩ ±3% @ 77°F (25°C)
- Beta Thermistor : 3960 K ±1%

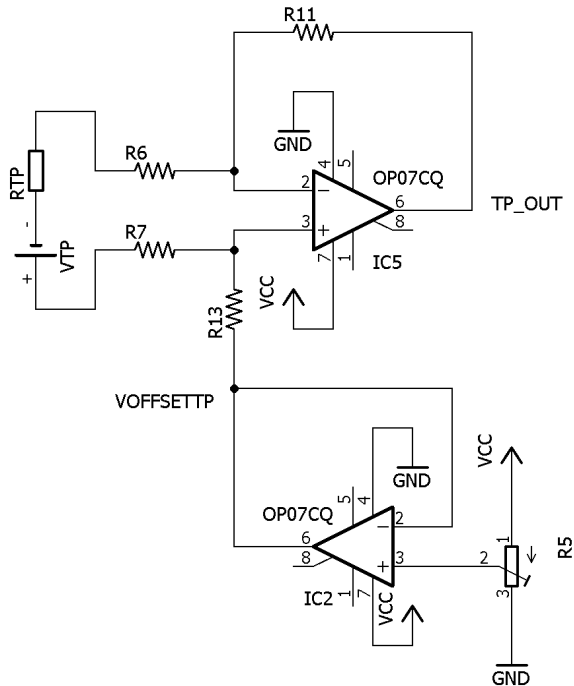
**Tabel 3.4.** Hubungan Nilai Resistansi Thermistor dengan Suhu

$T_{amb}$ (°C)	R (kΩ)	$T_{amb}$ (°C)	R (kΩ)
-20	947,900	40	53,150
-15	715,900	45	43,510
-10	545,400	50	35,790
-5	418,800	55	29,580
0	324,100	65	20,470
5	252,700	70	17,140
10	198,500	75	14,400
15	156,900	80	12,150
20	124,900	85	10,290
25	100,000	90	8,745
30	80,550	95	7,460
35	65,250	100	6,388

Tabel 3.4. dan Gambar 3.15. di bawah ini adalah spesifikasi hubungan nilai resistansi (R) thermistor dalam thermopile ZTP-135SR terhadap suhu ( $T_{amb}$ ):



**Gambar 3.15.** Fungsi Nilai Resistansi Thermistor dengan Suhu



**Gambar 3.16.** Rangkaian Pengondisi Sinyal Thermopile

### 3.6.1. Perangkat Keras Sensor Inframerah

#### 3.6.1.1. Perangkat Keras Thermopile

Rangkaian pengondisi sinyal thermopile pada Gambar 3.16 menggunakan rangkaian penguat diferensial dengan penambahan tegangan *offset* ( $V_{OFF\_TP}$ ) dari keluaran rangkaian *non-inverting voltage follower*. Resistor variabel  $R_5$  digunakan untuk mengatur tegangan output *voltage follower* dengan dasar rangkaian pembagi tegangan. Nilai-nilai resistor pada rangkaian pengondisi sinyal thermopile antara lain:

- $R_6 = 1 \text{ k}\Omega$
- $R_7 = 1 \text{ k}\Omega$
- $R_{11} = 10 \text{ M}\Omega$
- $R_{13} = 10 \text{ M}\Omega$

Dengan menggunakan persamaan (2.22) dan (2.18), maka penguatan ( $A_V$ ) yang dilakukan oleh rangkaian pengondisi sinyal thermopile adalah sebagai berikut:

$$\begin{aligned}
 V_{OUT} &= \frac{R_{11}}{R_6} (V_2 - V_1) + V_{OFF\_TP} \\
 V_{OUT} &= \frac{10 \text{ M}\Omega}{1 \text{ k}\Omega} * V_{TP} + V_{OFF\_TP} \\
 A_V &= \frac{V_{OUT} - V_{OFF\_TP}}{V_{IN}} \\
 A_V &= \frac{V_{OUT} - V_{OFF\_TP}}{V_{TP}} = 10000 \\
 V_{TP} &= \frac{V_{OUT} - V_{OFF\_TP}}{10000}
 \end{aligned} \tag{3.1}$$

### 3.6.1.2. Perangkat Keras Thermistor

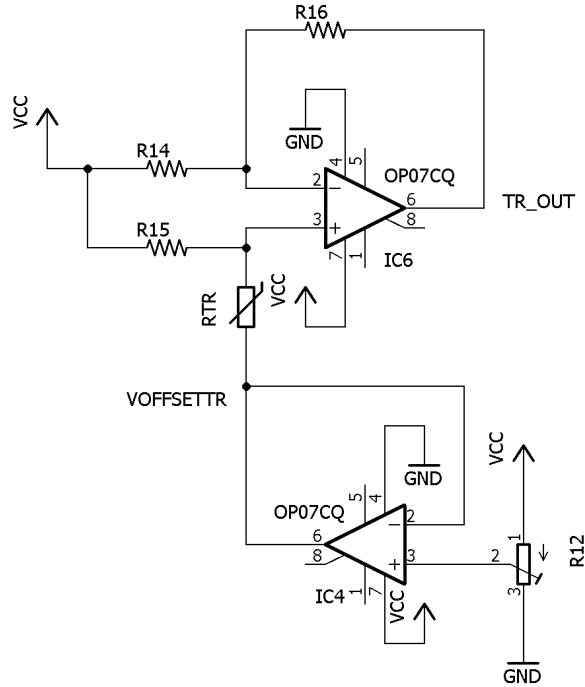
Rangkaian pengondisi sinyal thermistor pada Gambar 3.17 menggunakan rangkaian penguat diferensial dengan penambahan tegangan *offset* ( $V_{OFF\_TR}$ ) dari keluaran rangkaian *non-inverting voltage follower*. Resistor variable  $R_{13}$  digunakan untuk mengatur tegangan output *voltage follower* dengan dasar rangkaian pembagi tegangan.

Nilai-nilai resistor pada rangkaian pengondisi sinyal thermistor antara lain:

- $R_{14} = 22 \text{ k}\Omega$
- $R_{15} = 22 \text{ k}\Omega$
- $R_{16} = 33 \text{ k}\Omega$
- $R_{TR}$  = nilai resistansi thermistor terhadap suhu

Dengan menggunakan persamaan (2.21), maka tegangan keluaran ( $V_{out}$ ) dari rangkaian pengondisi sinyal thermistor adalah sebagai berikut:

$$\begin{aligned}
 V_{out} &= -V_{CC} \left( \frac{R_{16}}{R_{14}} \right) + V_{CC} \left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) \left( \frac{R_{14} + R_{16}}{R_{14}} \right) + V_{OFF\_TR} \\
 V_{out} &= V_{CC} \left[ \left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) \left( \frac{R_{14} + R_{16}}{R_{14}} \right) - \left( \frac{R_{16}}{R_{14}} \right) \right] + V_{OFF\_TR}
 \end{aligned} \tag{3.2}$$



**Gambar 3.17.** Rangkaian Pengondisi Sinyal Thermistor

### 3.6.2. Perangkat Lunak Sensor Inframerah

Untuk menghitung nilai suhu objek yang akan diukur, digunakan persamaan (2.8)

$$T_b \approx T_{sc} + \frac{V}{4Ag\epsilon\epsilon_s\sigma T_{sc}^3}$$

$$T_b \approx T_{sc} + \frac{V}{4 \cdot K_P \cdot T_{sc}^3} \quad (3.3)$$

Di mana:

$T_b$  : Suhu objek (K)

$T_{sc}$  : Suhu sensor (K), dapat diperoleh dengan komputasi dari nilai resistansi thermistor dari tegangan output rangkaian pengondisi sinyal thermistor

$V$  : Tegangan kaluaran sensor (V), yaitu yang diperoleh dari tegangan keluaran rangkaian pengondisi sinyal thermopile

$A$ : Luas permukaan sensor ( $m^2$ )	
$g$ : Responsivitas ( $V/W$ )	“Faktor Kalibrasi”,
$\varepsilon$ : Emisivitas objek	nilainya tidak berubah
$\varepsilon_s$ : Emisivitas sensor	} $K_p$ : pada setiap pengukuran, yang diperoleh pada saat kalibrasi
$\sigma$ : Konstanta Stefan-Boltzman ( $5,67 \times 10^{-8} W.m^{-2}.K^{-4}$ )	

a. Menghitung suhu sensor dengan thermistor

Suhu sensor dapat dihitung dengan persamaan (2.13):

$$T = [A + B(\ln S) + C(\ln S)^3]^{-1}$$

$$T_{sc} = [A + B(\ln R_{TR}) + C(\ln R_{TR})^3]^{-1} \quad (3.4)$$

Nilai koefisien  $A$ ,  $B$ , dan  $C$  dapat dihitung dengan persamaan (2.14), dengan nilai sesuai spesifikasi thermistor pada Tabel 3.4:

- $R_1 = 324,1 \text{ k}\Omega$  pada  $T_1 = 0^\circ\text{C} = 273 \text{ K}$ ,
- $R_2 = 80,5 \text{ k}\Omega$  pada  $T_2 = 30^\circ\text{C} = 303 \text{ K}$ ,
- $R_3 = 6,388 \text{ k}\Omega$  pada  $T_3 = 100^\circ\text{C} = 373 \text{ K}$

$$L_1 = \ln(R_1) = 12,689 \quad Y_1 = \frac{1}{T_1} = 3,663 \times 10^{-3}$$

$$L_2 = \ln(R_2) = 11,297 \quad Y_2 = \frac{1}{T_2} = 3,300 \times 10^{-3}$$

$$L_3 = \ln(R_3) = 8,762 \quad Y_3 = \frac{1}{T_3} = 2,681 \times 10^{-3}$$

$$\gamma_2 = \frac{Y_2 - Y_1}{L_2 - L_1} = 260,509 \times 10^{-6}$$

$$\gamma_3 = \frac{Y_3 - Y_1}{L_3 - L_1} = 250,097 \times 10^{-6}$$

$$C = \left( \frac{\gamma_3 - \gamma_2}{L_3 - L_2} \right) (L_1 + L_2 + L_3)^{-1} = 125,448 \times 10^{-9}$$

$$B = \gamma_2 - C(L_1^2 + L_1 L_2 + L_2^2) = 206,32 \times 10^{-6}$$

$$A = Y_1 - (B + L_1^2 C) L_1 = 788,759 \times 10^{-6}$$

Sedangkan nilai  $R_{TR}$  dapat diperoleh dari tegangan keluaran ( $V_{out}$ ) rangkaian pengondisi sinyal thermistor dengan membalik persamaan (3.2):

$$\begin{aligned}
V_{out} &= V_{CC} \left[ \left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) \left( \frac{R_{14} + R_{16}}{R_{14}} \right) - \left( \frac{R_{16}}{R_{14}} \right) \right] + V_{OFF\_TR} \\
\frac{V_{out} - V_{OFF\_TR}}{V_{CC}} &= \left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) \left( \frac{R_{14} + R_{16}}{R_{14}} \right) - \left( \frac{R_{16}}{R_{14}} \right) \\
\left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) \left( \frac{R_{14} + R_{16}}{R_{14}} \right) &= \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \\
\left( \frac{R_{TR}}{R_{15} + R_{TR}} \right) &= \left[ \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \right] \left( \frac{R_{14}}{R_{14} + R_{16}} \right) \\
\left( \frac{1}{\frac{R_{15} + R_{TR}}{R_{TR}}} \right) &= \left[ \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \right] \left( \frac{R_{14}}{R_{14} + R_{16}} \right) \\
\left( \frac{1}{\frac{R_{15}}{R_{TR}} + 1} \right) &= \left[ \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \right] \left( \frac{R_{14}}{R_{14} + R_{16}} \right) \\
\text{substitusi} \quad \left[ \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \right] \left( \frac{R_{14}}{R_{14} + R_{16}} \right) &= x \quad , \text{ sehingga:}
\end{aligned}$$

$$x = \left( \frac{1}{\frac{R_{15}}{R_{TR}} + 1} \right)$$

$$x \left( \frac{R_{15}}{R_{TR}} + 1 \right) = 1$$

$$\left( \frac{R_{15}}{R_{TR}} \right) x + x = 1$$

$$\left( \frac{R_{15}}{R_{TR}} \right) x = 1 - x$$

$$R_{TR} = \frac{x - R_{15}}{1 - x}$$

Jadi, nilai  $R_{TR}$  adalah  $R_{TR} = \frac{x - R_{15}}{1 - x}$   
dengan  $x = \left[ \frac{V_{out} - V_{OFF\_TR}}{V_{CC}} + \left( \frac{R_{16}}{R_{14}} \right) \right] \left( \frac{R_{14}}{R_{14} + R_{16}} \right)$  (3.5)

Dari persamaan (3.5) dan (3.2) dapat ditulis kode program untuk menghitung suhu thermistor:

```
const int trPin = A2; //pin ADC thermistor
float trADC = 0; //nilai ADC thermistor
//persamaan thermistor modelsteihart-hart
float trA = 788.759E-6, trB = 206.32E-6, trC = 125.448E-9;
//rangkaiian pengondisi sinyal thermistor
float trVcc = 5, trR8 = 22000, trR9 = 22000, trR10 = 33000;
float trVoff = 2.46, trVout, trX, trRTr;
float trSuhu; //suhu thermistor
//UKUR SUHU SENSOR
trADC = analogRead(trPin);
//tegangan output rangkaian pengondisi sinyal thermistor
trVout = trADC*5/1024;
//menghitung nilai resistansi thermistor
trX = (((trVout-trVoff)/trVcc)+
(trR10/trR8))*(trR8/(trR8+trR10));
trRTr = (trX*trR9)/(1-trX);
//menghitung suhu sensor dengan thermistor
trSuhu = 1/(trA+trB*log(trRTr)+trC*pow(log(trRTr),3));
```

b. Kalibrasi sensor thermopile untuk mendapatkan “faktor kalibrasi”

Untuk mendapatkan faktor kalibrasi, persamaan (3.3) dibalik menjadi:

$$T_b \approx T_{sc} + \frac{V}{4K_P T_{sc}^3}$$

$$K_P \approx \frac{V}{4 \cdot T_{sc}^3 (T_B - T_{sc})} \quad (3.6)$$

Hasil dari pengujian kalibrasi ini ada pada Sub subbab 4.1.1. dari pengujian yang dilakukan, didapatkan nilai dari faktor kalibrasi ( $K_P$ ) =  $2,65 \times 10^{-15}$

c. Menghitung suhu objek

Perhitungan suhu objek menggunakan persamaan (3.3):

$$T_b \approx T_{sc} + \frac{V}{4 \times 2,65 \times 10^{-15} \times T_{sc}^3}$$

Di mana:

$$V = V_{TP} = \frac{V_{OUT} - V_{OFF\_TP}}{10000}$$



Sehingga kode program untuk mengukur suhu objek adalah sebagai berikut:

```
const int tpPin = A3;  
float tpVout, tpVtp, tpVoff=2.15, tpADC;  
float suhuObjek;  
tpADC = analogRead(tpPin);  
tpVout = tpADC*5/1024;  
tpVtp = (tpVout-tpToff)/10000;  
suhuObjek = trSuhu + tpVtp/(4*2.65E-15*pow(trSuhu,3));
```

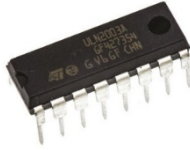
### 3.7. Perancangan Penggerak Motor Stepper

Motor stepper yang digunakan adalah 28BYJ-48 5V dengan bentuk fisik seperti pada Gambar 3.18. Motor stepper ini memiliki spesifikasi sebagai berikut:

- Tegangan : 5 V
- Jumlah fase : 4
- Rasio variasi kecepatan : 1/64
- Sudut langkah (*step*) : 5,625°/64
- Frekuensi : 100 Hz
- Resistansi DC :  $50 \Omega \pm 7\%$  (25°C)
- Frekuensi *idle in-traction* : >600 Hz
- Frekuensi *idle out-traction* : >1000 Hz
- Torsi *in-traction* : >34,3 mN.m (120 Hz)
- Torsi *self-positioning* : >34,3 mN.m
- Torsi *friction* : 600-1200 gf.cm
- Torsi *pull-in* : 300 gf.cm
- Resistansi *insulated* : >10 M $\Omega$  (500 V)
- Daya listrik *insulated* : 600 VAC/1 mA/ 1 s



**Gambar 3.18.** Bentuk Fisik Motor Stepper 28BYJ-48 5V



**Gambar 3.19.** Bentuk Fisik ULN2003A

Untuk membantu menggerakkan motor stepper, digunakan IC susunan transistor Darlington ULN2003A seperti pada Gambar 3.19. Perangkat ULN2003A adalah susunan transistor Darlington bertegangan tinggi dan arus tinggi. Masing-masing terdiri dari tujuh pasang NPN Darlington yang menonjolkan output tegangan tinggi dengan clamp-diode *common cathode* untuk *switching* beban induktif. Rating arus kolektor sepasang Darlington tunggal adalah 500 mA. Pasangan Darlington dapat diparalelkan untuk kemampuan arus yang lebih tinggi.

### 3.7.1. Perangkat Keras Penggerak Motor Stepper

Hubungan antara pin Arduino, ULN2003A, dan motor stepper ditunjukkan pada Gambar 3.20. Dengan suplai tegangan 5 volt dari keluaran rangkaian regulator catu daya.

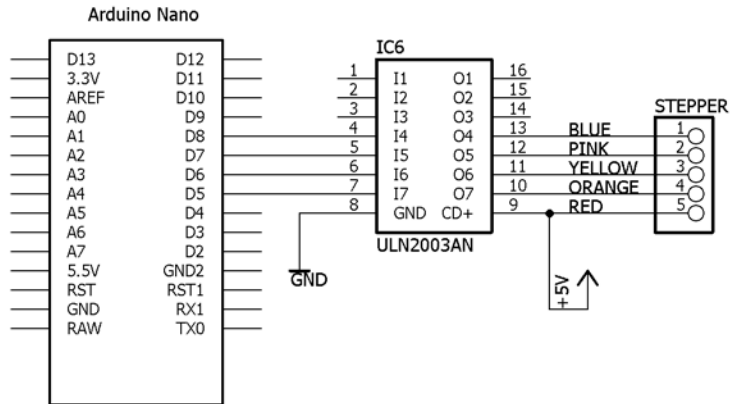
### 3.7.2. Perangkat Lunak Penggerak Motor Stepper

Pemrograman untuk memutar motor stepper pada Arduino Nano menggunakan fungsi pada *library* Stepper.h.

```
#include <Stepper.h>
//jumlah step dalam satu revolusi motor internal dalam mode full-
  step (4 step)
#define STEPS_PER_MOTOR_REVOLUTION 32
//gear-ratio 64:1 sehingga 32*64 step dalam satu revolusi shaft
  output (yang sebenarnya 63.68395:1)
#define STEPS_PER_OUTPUT_REVOLUTION 32 * 64 //2048 (2038)

Stepper smallStepper(STEPS_PER_MOTOR_REVOLUTION, 8, 6, 7, 5);
int stepStepper; //deklarasi stepper dengan urutan jumlah step
  per satu revolusi shaft, In1, In3, In2, In4

void putarStepper(int stepStepper){
  smallStepper.setSpeed(700);
  smallStepper.step(stepStepper);
}
```



**Gambar 3.20.** Skematik Rangkaian Penggerak Motor Stepper

### 3.8. Perancangan Penggerak Motor Servo

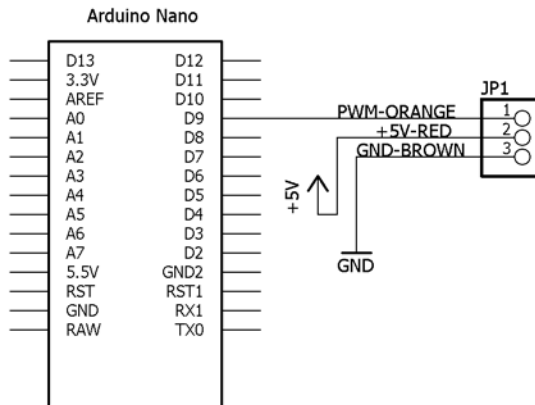
Motor servo yang digunakan adalah TowerPro MG995 seperti pada Gambar 3.21. Motor servo ini memiliki spesifikasi sebagai berikut:

- Berat : 55 gram
- Dimensi : 40,7 x 19,7 x 42,9 mm
- Torsi : 8,5 kgf.cm (4,8 V), 10 kgf.cm (6 V)
- Kecepatan : 0,2 s/60° (4,8 V), 0,16 s/60° (6 V)
- Tegangan : 4,8 V s.d. 7,2 V
- *Dead-band width* : 5  $\mu$ s
- Rentang suhu : 0°C s.d. 55°C



**Gambar 3.21.** Bentuk Fisik Motor Servo MG995

### 3.8.1. Perangkat Keras Penggerak Motor Servo



**Gambar 3.22.** Skematik Rangkaian Penggerak Motor Servo

Hubungan antara pin Arduino dan motor servo ditunjukkan pada Gambar 3.22. Dengan suplai tegangan 5 volt dari keluaran rangkaian regulator catu daya.

### 3.8.2. Perangkat Lunak Penggerak Motor Servo

Pemrograman motor servo perlu fungsi-fungsi dalam *library* Servo.h.

```
#include <Servo.h>
```

```
Servo myservo; //definisi nama struct servo
```

```
myservo.attach(9); //pin PWM motor servo
```

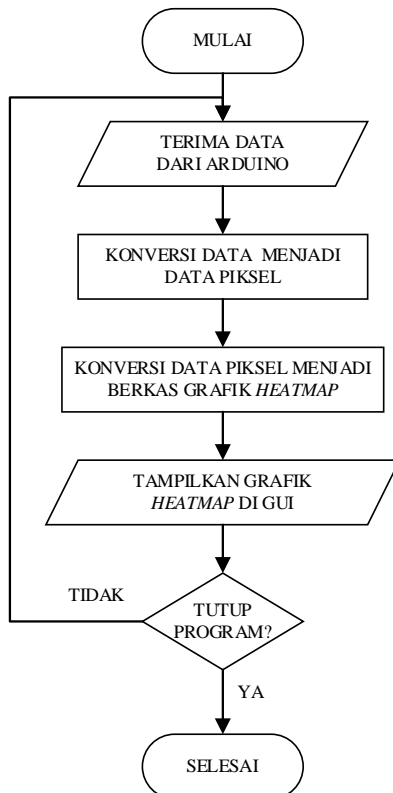
```
myservo.write(pos); //sudut pos bernilai 0 s.d. 180 derajat
```

```
delay(20); //menunggu 20 ms untuk servo mencapai sudutnya
```

## 3.9. Perangkat Lunak Pemetaan Suhu

Pemrograman di komputer dengan menggunakan IDE Microsoft Visual Studio bertujuan untuk mendapatkan GUI (*Graphic User Interface*). GUI ini dirancang untuk menampilkan visualisasi pemetaan data hasil pengukuran suhu. Data ukur suhu dikirim dari Arduino melalui komunikasi serial dan diterima oleh port serial pada komputer.

Pemrograman dalam Microsoft Visual Studio menggunakan bahasa pemrograman C++. Bahasa C++ dalam Visual Studio menggunakan *managed C++*. Kebanyakan *syntax managed C++* sama dengan *unmanaged C++* namun ada perbedaan sedikit. Untuk merancang sebuah GUI diperlukan fasilitas Windows Form. Untuk merancang Windows Form berbasis C++ menggunakan *CLR project*. Diagram alir pemrograman peemetaan ini ditunjukkan pada Gambar 3.23.



**Gambar 3.23.** Diagram Alir Pemrograman Pemetaan

### 3.9.1. Fasilitas Port Serial

Fasilitas port serial digunakan untuk menerima data ukur suhu dari Arduino. Setelah diterima melalui serial, data akan disimpan dalam *array* menggunakan *pointer*.

```
serialPort1->Open();

terimaSerialManaged = serialPort1->ReadLine();

delimiter = delimStr->ToCharArray();
words = terimaSerialManaged->Split(delimiter);

if (words->Length > 2) {
    dataCacahMiring = (int)(Convert::ToInt16(serialDatas[0]));
    dataCacahSudut = (int)(Convert::ToInt16(serialDatas[1]));
    dataMiring = (float)(Convert::ToDouble(serialDatas[2]));
    dataSudut = (float)(Convert::ToDouble(serialDatas[3]));
    dataSuhu = (float)(Convert::ToDouble(serialDatas[4]));
    *(miring + jumSpot*dataCacahMiring + dataCacahSudut) =
        dataMiring;
    *(kompas + jumSpot*dataCacahMiring + dataCacahSudut) =
        dataSudut;
    *(suhu + jumSpot*dataCacahMiring + dataCacahSudut) = dataSuhu;
}
```

### 3.9.2. Konversi Data Suhu ke Data Pixel

Data suhu yang tersimpan kemudian diproses untuk menghasilkan data piksel. Data piksel ini berisi nilai RGBA (pewarnaan) yang nilai nya proporsional terhadap data suhu pada suatu titik. Konversi dari data suhu menjadi data piksel ini menggunakan *library* heatmap.h.

Pertama deklarasi struct heatmap\_t yang bernama hm dengan fungsi heatmap\_new dengan lebar w piksel dan tinggi h piksel. Kemudian digunakan fungsi heatmap\_stamp\_gen\_nonlinear untuk menghasilkan pola dari setiap data ukur dengan radius pola sebesar rStamp. Data ukur suhu yang tersimpan kemudian diberi pola dan diletakkan sesuai koordinat piksel. Pada akhirnya, setiap data piksel dikonversi ke nilai RGBA sesuai dengan skema pewarnaan.

```
heatmap_t* hm = heatmap_new(w, h);
auto pola = heatmap_stamp_gen_nonlinear(rStamp);

for (int cacahRadius = 0; cacahRadius < jumRadius; cacahRadius++)
{
    for (int cacahSpot = 0; cacahSpot < jumSpot; cacahSpot++) {
```

```

weight = *(suhu + jumSpot*cacahRadius + cacahSpot);
pikselStepMiring = *(miring + jumSpot*cacahRadius +
    cacahSpot);
pikselStepSudut = *(kompas + jumSpot*cacahRadius + cacahSpot);

pikselRadius = ((w / 2) - (rStamp / 4))
    *(tan(pikselStepMiring*PI/180) / tan(miringMax*PI / 180));

x = w / 2 + pikselRadius*sin(pikselStepSudut*PI / 180);
y = h / 2 - pikselRadius*cos(pikselStepSudut*PI / 180);

heatmap_add_weighted_point_with_stamp(hm, x, y, weight, pola);
}
}

std::vector<unsigned char> image(w*h * 4);
heatmap_render_default_to(hm, &image[0]);

heatmap_free(hm);

```

### 3.9.3. Menghasilkan Berkas Gambar dari Data Pixel

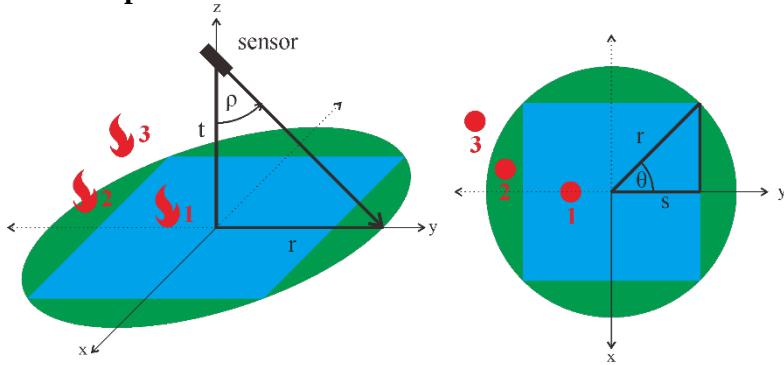
Data piksel yang sudah bernilai RGBA kemudian dikonversi menjadi berkas gambar berekstensi \*.png dengan fungsi pada *library* `lodepng.h`.

```

if (unsigned error = lodepng::encode("heatmap.png", image, w, h))
{
    std::cerr << "encoder error " << error << ": " <<
        lodepng_error_text(error) << std::endl;
}

```

### 3.10. Aplikasi Praktis Sistem Pemetaan



**Gambar 3.24.** Aplikasi Praktis Sistem Pemetaan

Keterangan Gambar 3.24.

- $r$  : Radius maksimum pemetaan dari putaran motor servo
- $s$  : Setengah sisi pemetaan GUI
- $t$  : Tinggi peletakan sensor inframerah
- $\rho$  : Kemiringan maksimum sensor inframerah
- $\theta$  :  $45^\circ$

Aplikasi praktis dari sistem pemetaan suhu lahan ini dengan meletakkan sistem di tengah-tengah lahan yang akan dipetakan. Gambar 3.24. mengilustrasikan area lahan berbentuk lingkaran yang terukur suhunya pada warna hijau dan biru. Sedangkan hanya area lahan berwarna biru berbentuk persegi yang ditampilkan pada GUI pemetaan di komputer. Jika terdapat objek bersuhu tinggi / kebakaran di titik 1, maka akan tertampil di GUI pemetaan. Jika terdapat objek bersuhu tinggi / kebakaran di titik 2, maka tidak akan tertampil di GUI pemetaan walaupun terukur oleh sensor. Jika terdapat objek bersuhu tinggi / kebakaran di titik 3, maka tidak terukur oleh sensor sehingga tidak akan tertampil di GUI pemetaan.

$$\begin{aligned}
 r &= t \times \tan \rho \\
 s &= r \times \cos \theta \\
 s &= t \times \tan \rho \times \cos \theta
 \end{aligned}
 \tag{3.7}$$



Dengan peletakan sensor pada ketinggian  $t$  dan kemiringan maksimum  $\rho$  yang diatur oleh motor servo, sehingga luas area lahan yang terpetakan pada program GUI adalah  $2s \times 2s$  sesuai pada persamaan (3.7). Hubungan tinggi peletakan sensor, kemiringan maksimum dan luas area lahan yang terpetakan ditunjukkan pada Tabel 3.5. Dengan tinggi sensor 3 meter dan kemiringan  $45^\circ$  maka luas area yang terpetakan adalah  $4,24 \times 4,24 = 18 \text{ m}^2$ . Dengan tinggi sensor 10 meter dan kemiringan  $45^\circ$  maka luas area yang terpetakan adalah  $14,14 \times 14,14 = 200 \text{ m}^2$ .

Untuk alternatif pemetaan dengan luas lahan yang lebih besar, kemiringan sensor yang diatur oleh motor servo dapat diperbesar, misal  $60^\circ$  atau  $75^\circ$ . Dengan tinggi sensor 3 meter dan kemiringan  $60^\circ$  maka luas area yang terpetakan adalah  $7,35 \times 7,35 = 54 \text{ m}^2$ . Dengan tinggi sensor 10 meter dan kemiringan  $60^\circ$  maka luas area yang terpetakan adalah  $24,49 \times 24,49 = 600 \text{ m}^2$ . Dengan tinggi sensor 3 meter dan kemiringan  $75^\circ$  maka luas area yang terpetakan adalah  $15,83 \times 15,83 = 250,71 \text{ m}^2$ . Dengan tinggi sensor 10 meter dan kemiringan  $75^\circ$  maka luas area yang terpetakan adalah  $52,78 \times 52,78 = 2785,70 \text{ m}^2$ .

**Tabel 3.5.** Aplikasi Praktis Sistem Pemetaan

T (m)	$\rho$ ( $^\circ$ )	s (m)	Sisi (m)	Luas ( $\text{m}^2$ )
3	45	2,12	4,24	18,00
5	45	3,54	7,07	50,00
7	45	4,95	9,90	98,00
10	45	7,07	14,14	200,00
3	60	3,67	7,35	54,00
5	60	6,12	12,25	150,00
7	60	8,57	17,15	294,00
10	60	12,25	24,49	600,00
3	75	7,92	15,83	250,71
5	75	13,19	26,39	696,42
7	75	18,47	36,95	1364,99
10	75	26,39	52,78	2785,70

Untuk pemetaan suhu permukaan lahan yang lebih luas digunakan *drone* untuk menempatkan sistem pemetaan ini. Dengan ketinggian maksimum *uncontrolled airspace* yang diperbolehkan adalah 150 meter dan kemiringan maksimum  $45^\circ$  sensor inframerah, sesuai persamaan (3.7) luas area yang dapat teramati adalah  $45.000 \text{ m}^2$ . Untuk pengamatan yang lebih luas, pengamatan dilakukan pada beberapa titik *waypoint* GPS pada *drone* yang diatur lebih dulu.

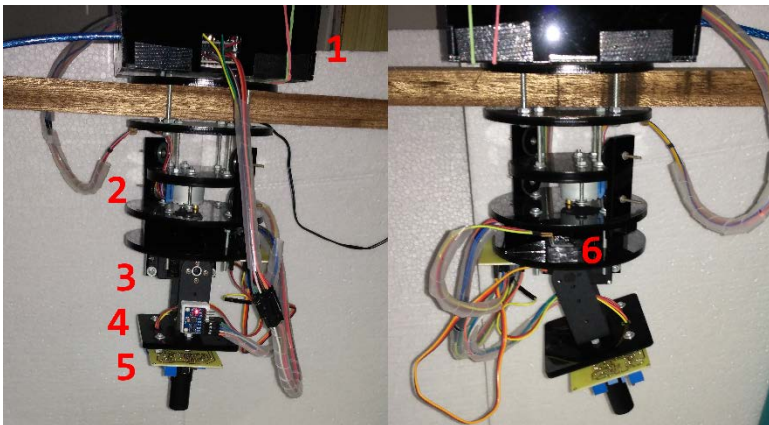
## **BAB IV**

### **PENGUJIAN DAN PEMBAHASAN SISTEM**

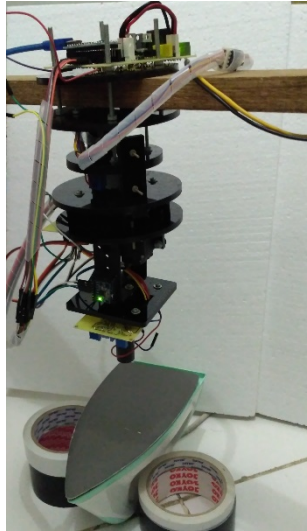
Sistem yang dirancang yang ditunjukkan pada Gambar 4.1. memiliki bagian utama yang mengandung rangkaian regulator dan Arduino Nano pada nomor 1, motor servo MG995 pada nomor 3 dan motor stepper 28BYJ-48 pada nomor 2 sebagai pengubah arah pengukuran suhu, sensor inframerah thermopile dan rangkaian pengondisi sinyal pada nomor 5, sensor akselerometer MPU-6050 pada nomor 4, dan sensor magnetometer HMC5983 pada nomor 6.

#### **4.1. Pengujian Rangkaian Sensor Thermopile ZTP-135SR**

Pengujian dilakukan dengan mengukur suhu objek berupa benda panas yang diatur agar suhu menurun dengan berjalannya waktu seperti pada Gambar 4.2. Dalam pengujian ini, setrika listrik digunakan dengan diatur untuk menghasilkan panas maksimal, setelah itu sumber tegangan dimatikan. Setelah sumber tegangan tegangan tersebut dimatikan, panas yang dihasilkan setrika listrik akan terus turun begitu juga suhunya. Jarak antara sensor dan sumber panas yaitu sekitar 20 cm.



**Gambar 4.1.** Bentuk Fisik dari Sistem



**Gambar 4.2.** Pengujian Sensor Thermopile

#### 4.1.1. Pengujian Kalibrasi untuk Mendapatkan $K_P$

Untuk menghitung nilai faktor kalibrasi ( $K_P$ ) dapat digunakan persamaan (3.1) dan (3.6):

$$K_P \approx \frac{V}{4 \cdot T_{SC}^3 (T_B - T_{SC})}$$

Di mana:

$$V = V_{TP} = \frac{V_{OUT} - V_{OFF\_TP}}{10000}$$

$V_{TP}$ : tegangan yang dihasilkan thermopile (volt)

$V_{OUT}$ : tegangan keluaran rangkaian pengondisi sinyal thermopile (volt)

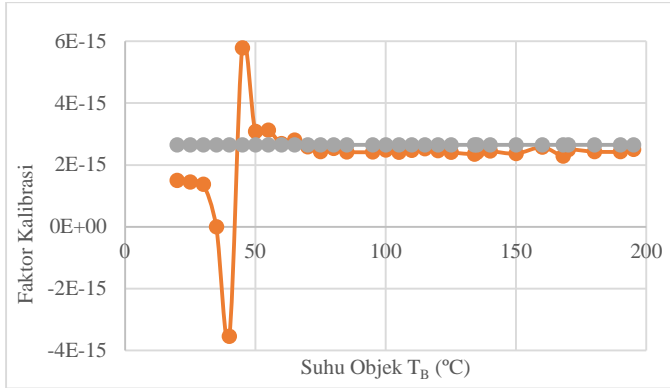
$V_{OFF\_TP}$ : Tegangan *offset* pada rangkaian pengondisi sinyal thermopile (volt)

$T_{SC}$ : Suhu sensor yang diukur dengan thermistor (K)

$T_B$ : Suhu objek yang diukur eksternal dengan thermogun (K)

**Tabel 4.1.** Pengujian Untuk Mendapatkan Faktor Kalibrasi

$V_{TP}$ (mV)	$T_{SC}$ (K)	$T_B$ (°C)	$T_B$ (K)	$K_F (\times 10^{-15})$
0,048	315,3961	195	468,15	2,5039
0,045	315,3961	190	463,15	2,4269
0,042	315,3961	180	453,15	2,4295
0,040	315,3961	170	443,15	2,4949
0,036	315,3961	168	441,15	2,2811
0,038	315,3961	160	433,15	2,5715
0,032	315,3961	150	423,15	2,3664
0,030	315,3961	140	413,15	2,4454
0,028	315,3961	135	408,15	2,4055
0,024	315,3961	120	393,15	2,4596
0,021	315,3961	110	383,15	2,4698
0,018	315,3961	100	373,15	2,4835
0,016	315,3961	95	368,15	2,4168
0,012	315,3961	80	353,15	2,5327
0,010	315,3961	75	348,15	2,4328
0,009	315,3961	70	343,15	2,584
0,008	315,3961	65	338,15	2,8016
0,006	315,3961	60	333,15	2,693
0,005	315,3961	55	328,15	3,1239
0,003	315,3961	50	323,15	3,083
0,002	315,3961	45	318,15	5,7871
0,001	315,3961	40	313,15	-3,5476
0	315,1184	35	308,15	0
-0,002	314,8398	30	303,15	1,3705
-0,003	314,7468	25	298,15	1,4493
-0,004	314,6537	20	293,15	1,4928



**Gambar 4.3.** Nilai Faktor Kalibrasi Terhadap Suhu Objek Pengujian

Hasil pengujian ditunjukkan pada Tabel 4.1 dan Gambar 4.3 sebagai visualisasi. Dari hasil pengujian, didapatkan pada pengujian dengan suhu objek 50 °C s.d. 195 °C nilai  $K_p$  cukup konstan. Untuk mendapatkan nilai  $K_p$ , nilai  $K_p$  dari pengujian suhu objek 50 °C s.d. 195 °C dirata-ratakan sebagai berikut:

$$\overline{K_p} = \frac{1}{n} \sum_{i=1}^n K_{p_i} = 2.65 \times 10^{-15}$$

#### 4.1.2. Pengujian Pengukuran Suhu Objek

Pengukuran suhu objek dengan program dan rangkaian thermopile ( $T_{B(R)}$ ) dibandingkan dengan pengukuran secara manual menggunakan thermogun ( $T_{B(M)}$ ). Hasil pengujian ini ditunjukkan pada Tabel 4.2. Persentase kesalahan digambarkan dalam Gambar 4.4 dengan *error* rata-rata sebesar 6,4%. Pengujian sistem pengukur suhu ini mengukur suhu dengan pada rentang 37 °C s.d. 170 °C. Perhitungan *error* pengukuran dihitung dengan rumus sebagai berikut:

$$Error = \frac{T_{B(M)} - T_{B(R)}}{T_{B(M)}} \times 100\%$$

Tabel 4.2. Pengujian Pengukuran Suhu Objek

$T_{B(R)}$ (°C)	$T_{B(M)}$ (°C)	Error (%)
170	173	1,73
157	162	3,09
143	150	4,67
131	135	2,96
121	127	4,72
116	123	5,69
109	116	6,03
102	108	5,56
94	102	7,84
90	99	9,09
86	96	10,42
83	91	8,79
77	86	10,47

$T_{B(R)}$ (°C)	$T_{B(M)}$ (°C)	Error (%)
75	82	8,54
71	76	6,58
68	73	6,85
65	70	7,14
62	66	6,06
58	63	7,94
56	59	5,08
50	52	3,85
47	50	6,00
43	46	6,52
40	43	6,98
37	40	7,50

Pengujian berulang sebanyak 38 kali juga dilakukan kepada objek bersuhu konstan yang ditunjukkan pada Tabel 4.3 dan pada Gambar 4.4.

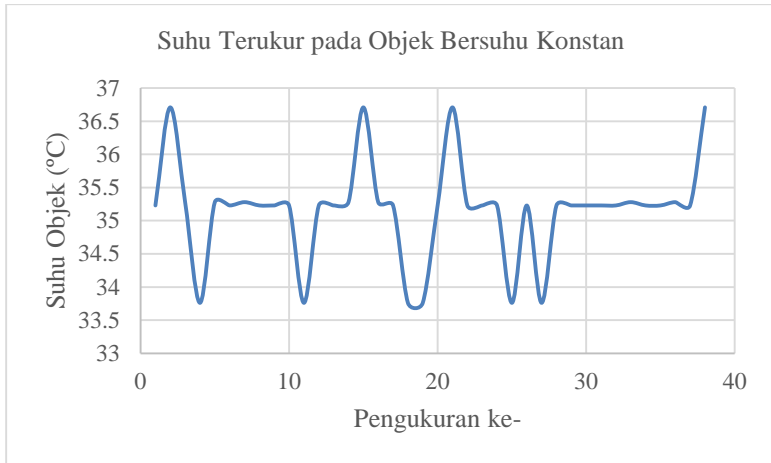


Gambar 4.4. Persentase Error pada Pengukuran Suhu Objek

**Tabel 4.3.** Pengujian pada Objek Bersuhu Konstan

Pengu- kuran ke-	T <sub>B</sub> (°C)	Pengu- kuran ke-	T <sub>B</sub> (°C)
1	35,23	20	35,23
2	36,71	21	36,71
3	35,28	22	35,23
4	33,76	23	35,23
5	35,28	24	35,23
6	35,23	25	33,76
7	35,28	26	35,23
8	35,23	27	33,76
9	35,23	28	35,23
10	35,23	29	35,23
11	33,76	30	35,23
12	35,23	31	35,23
13	35,23	32	35,23
14	35,28	33	35,28
15	36,71	34	35,23
16	35,28	35	35,23
17	35,23	36	35,28
18	33,76	37	35,23
19	33,76	38	36,71





**Gambar 4.5.** Pengujian pada Objek Bersuhu Konstan

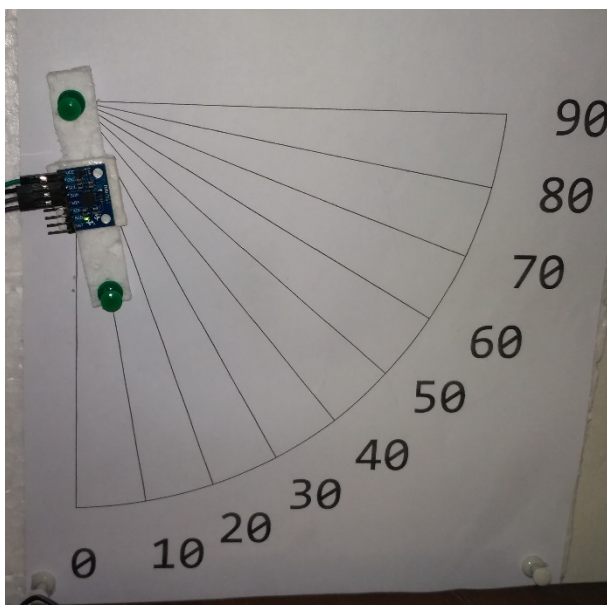
Dari Tabel 4.3 dan Gambar 4.4 didapatkan *error* yang terjadi ketika sistem mengukur objek bersuhu konstan sebesar  $\pm 1,5^{\circ}\text{C}$ .

## 4.2. Pengujian Sensor Akselerometer dan Motor Servo

### 4.2.1. Pengujian Sensor Akselerometer

Pengujian sensor akselerometer adalah dengan membaca nilai percepatan pada sumbu x, y, z, sudut kemiringan (*tilt*) yang terbaca, dan sudut kemiringan. Pengujian ini ditunjukkan pada Gambar 4.6 pada kemiringan  $10^{\circ}$ .

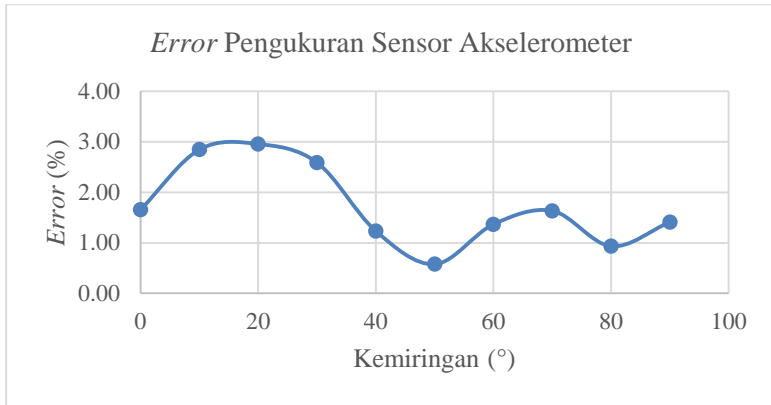
Dari hasil pengujian tersebut, didapatkan nilai derajat kemiringan terhadap sumbu y dari sensor akselerometer sesuai dengan kode program. Pada Tabel 4.4 menunjukkan nilai percepatan *Acc* dengan satuan g (percepatan gravitasi bumi  $9,8 \text{ m/s}^2$ ) pada setiap sumbu, kemiringan referensi *Tilt Ref*, kemiringan hasil pengukuran *Tilt Measured*, dan *error* yang dihasilkan. Pada Gambar 4.7 menunjukkan perbandingan *error* yang dihasilkan antara hasil pembacaan dengan kemiringan referensi. *Error* yang dihasilkan antara 0,58% s.d. 2,96% dengan rata-rata 1,72%.



**Gambar 4.6.** Pengujian Sensor Akselerometer

**Tabel 4.4.** Pengujian Sensor Akselerometer

<b>Tilt Ref</b>	<b>Tilt Measured</b>	<b>Acc X</b>	<b>Acc Y</b>	<b>Acc Z</b>	<b>Error (%)</b>
0	1,49	0,03	0,98	-0,06	1,66
10	12,56	0,22	0,97	-0,10	2,84
20	22,66	0,39	0,92	-0,11	2,96
30	32,33	0,54	0,86	-0,11	2,59
40	41,11	0,68	0,77	-0,10	1,23
50	49,48	0,78	0,67	-0,14	0,58
60	58,77	0,89	0,54	-0,10	1,37
70	68,53	0,98	0,38	-0,10	1,63
80	79,16	1,04	0,20	-0,10	0,93
90	88,73	1,07	0,02	-0,10	1,41



**Gambar 4.7.** Error Pengukuran Sensor Akselerometer

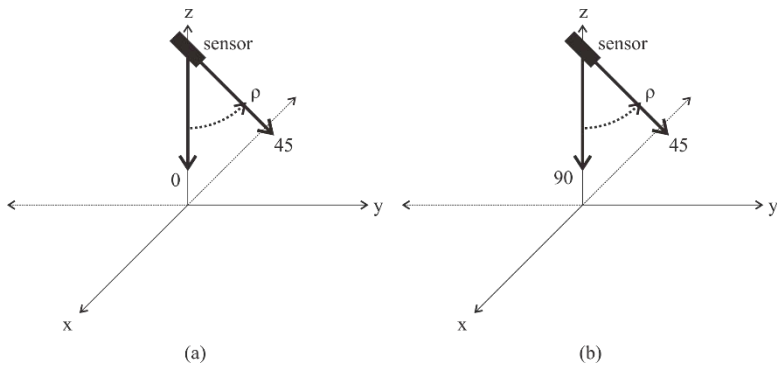
#### 4.2.2. Pengujian Motor Servo

Pengujian ini dilakukan untuk membandingkan kemiringan sensor inframerah yang dihasilkan dari pemrograman motor servo  $\rho_{calc}$  dengan hasil pengukuran dari akselerometer  $\rho_{acc}$ . Gambar 4.8. menunjukkan ilustrasi pemrograman motor servo terhadap kemiringan yang dihasilkan. Pada posisi motor servo bernilai  $90^\circ$  akan dihasilkan kemiringan sensor  $0^\circ$ . Sedangkan posisi motor servo bernilai  $45^\circ$  akan dihasilkan kemiringan sensor  $45^\circ$ . Dengan persamaan garis linier didapatkan hubungan tilt  $\rho_{calc}$  dengan nilai pada variabel  $pos$  untuk mengatur posisi putaran motor servo:

$$\rho_{calc} = 90 - pos$$

Program untuk pengujian ini adalah sebagai berikut:

```
for (posServo = 45; posServo <= 90; posServo += 5) {
  for(int i=0; i <10; i++)
  {
    myServo.write(posServo);
    delay(1000);
    for (i=0;i<10;i++){
      getAccel();
      Serial.print(posServo);Serial.print("\t");
      Serial.println(tiltFild);
    }
    delay(500);
  }
}
```

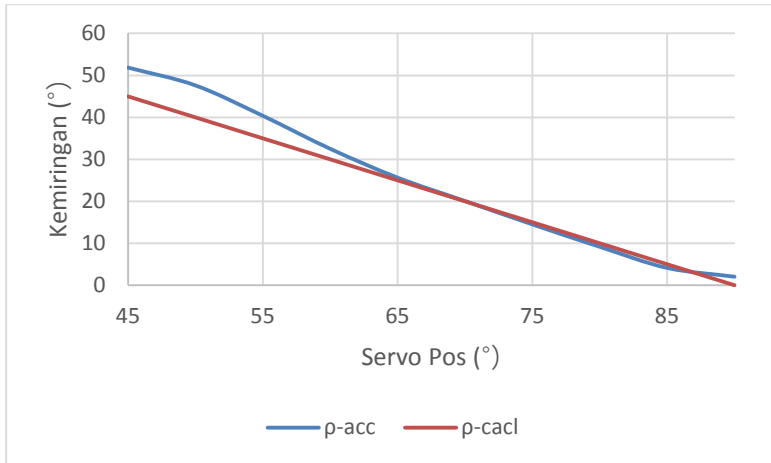


**Gambar 4.8.** Ilustrasi (a) Kemiringan dan (b) Putaran Servo

Pada Tabel 4.5. dan Gambar 4.8. menunjukkan perbandingan antara kemiringan yang dihasilkan dari pemrograman motor servo  $\rho_{calc}$  dengan hasil pengukuran dari akselerometer  $\rho_{acc}$ . Pada posisi servo  $65^\circ$  s.d.  $90^\circ$  dihasilkan *error* cukup kecil antara  $0,07^\circ$  s.d.  $2,03^\circ$ . Sedangkan pada posisi servo  $45^\circ$  s.d.  $60^\circ$  dihasilkan *error* yang sedang dari  $2,51^\circ$  s.d.  $7,62^\circ$ .

**Tabel 4.5.** Perbandingan Kemiringan Terukur dengan Servo

Servo pos ( $^\circ$ )	$\rho_{calc}$ ( $^\circ$ )	$\rho_{acc}$ ( $^\circ$ )	Error ( $^\circ$ )
45	45	51,87	6,87
50	40	47,62	7,62
55	35	40,38	5,38
60	30	32,51	2,51
65	25	25,64	0,64
70	20	20,07	0,07
75	15	14,50	0,50
80	10	9,16	0,84
85	5	4,13	0,87
90	0	2,03	2,03

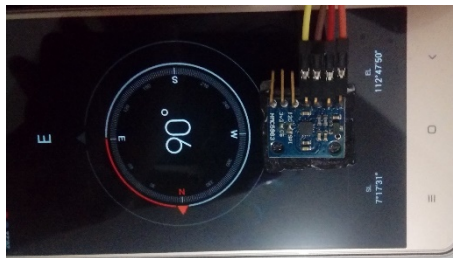


**Gambar 4.8.** Perbandingan Kemiringan Terukur dengan Servo

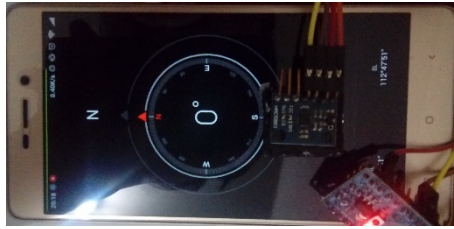
### 4.3. Pengujian Sensor Magnetometer dan Motor Stepper

#### 4.3.1. Pengujian Sensor Magnetometer

Pengujian sensor magnetometer adalah dengan membandingkan hasil pengukuran dan kode program dengan *software* kompas yang ada pada smartphone. Pengujian dilakukan dengan membandingkan arah mata angin 0° (360°), 45°, 90°, 135°, 180°, 225°, 270°, dan 315°. Pengujian ini ditunjukkan pada Gambar 4.9 dan 4.10.



**Gambar 4.9.** Pengujian Sensor Magnetometer dengan Arah 90°

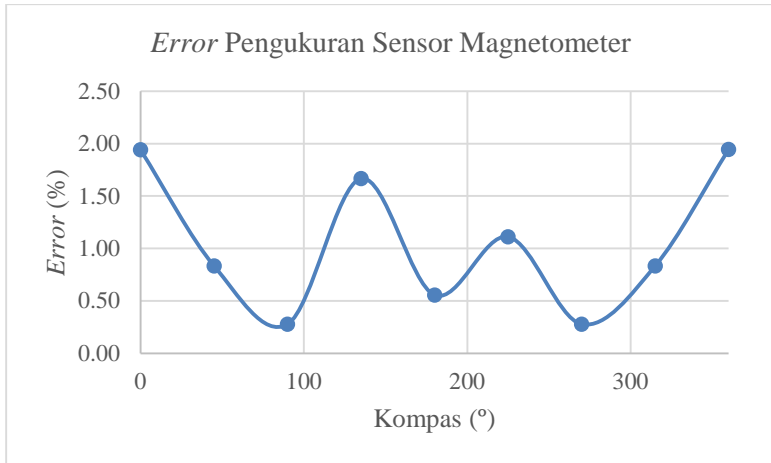


**Gambar 4.10.** Pengujian Sensor Magnetometer dengan Arah 0°

Dari hasil pengujian yang dilakukan, persentase *error* yang terjadi adalah antara 0,28% s.d. 1,94% seperti yang ditunjukkan pada Gambar 4.11 dan Tabel 4.6.

**Tabel 4.6.** Hasil Pengujian Sensor Magnetometer

Kompas (°)	Magneto meter (°)	<i>Error</i> (%)
45	42	0,83
90	89	0,28
135	141	1,67
180	182	0,56
225	229	1,11
270	269	0,28
315	318	0,83
360	353	1,94



**Gambar 4.11.** Error pada Pengujian Sensor Magnetometer

#### 4.3.2. Pengujian Motor Stepper

Dalam sistem ini, arah mata angin dari sistem sensor ditentukan oleh putaran motor stepper. Untuk itu putaran motor stepper perlu umpan balik dari pembacaan sensor magnetometer yang menghasilkan arah mata angin. Dalam pengujian ini dilakukan beberapa arah mata angin *set-point* yang harus ditunjuk sensor, yaitu arah  $0^\circ$ ,  $20^\circ$ ,  $40^\circ$ , dan  $60^\circ$ . Program yang digunakan sebagai berikut:

```
int stepperError, stepperCurrent;
int stepperSetPoint=b*20;
int magnHeadTop = stepperSetPoint + 8;
int magnHeadBottom = stepperSetPoint - 8;
Serial.print(b*20);Serial.print("\t");Serial.print(magnHeadTop);Serial.print("\t");Serial.print(magnHeadBottom);Serial.println("\t");
for (i=0; i<10; i++){
    getMagn();
    Serial.print("==");Serial.print("\t");Serial.print(magnHeadUnfild);Serial.print("\t");
    Serial.println(magnHeadFild[a][b]); delay(50);
}
while((magnHeadTop<magnHeadFild[a][b]) || (magnHeadBottom>magnHeadFild[a][b])){
    stepperCurrent = magnHeadFild[a][b];
    if ((stepperSetPoint>315) && (stepperCurrent<45))stepperCurrent+=360;
```

```

if ((stepperSetPoint<45) &&
    (stepperCurrent>225))stepperCurrent-=360;
stepperError=stepperSetPoint-stepperCurrent;
putarStepper(-(stepperError*5.7));
for (i=0; i<10; i++){
    getMagn();
    Serial.print("====");Serial.print("\t");Serial.print(magnH
        eadUnfilid);Serial.print("\t");
    Serial.println(magnHeadFild[a][b]);
    delay(50);
}
}

```

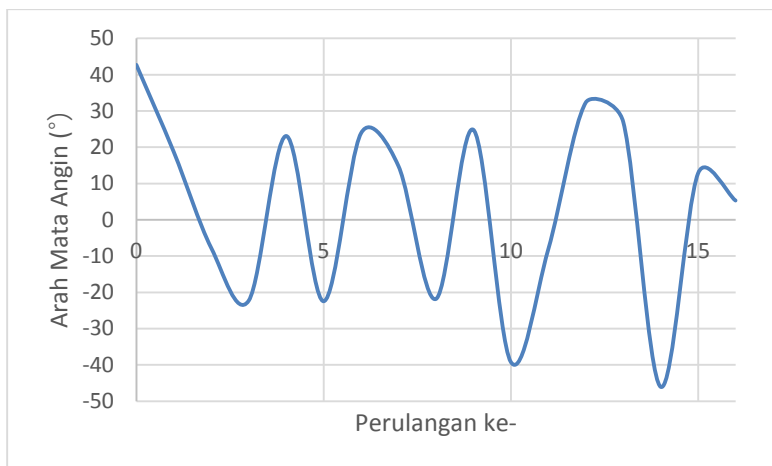
Dalam program ini dilakukan pembacaan arah mata angin dengan sensor magnetometer. Kemudian masuk ke perulangan *while*. Jika arah sensor inframerah bernilai *set-point*  $\pm 8^\circ$ , maka melakukan instruksi dalam perulangan, namun jika diluar tersebut akan melakukan instruksi perulangan *while* dengan menghitung nilai *error* yang dilanjutkan dengan perintah putar motor stepper sesuai dengan error yang terjadi. Nilai 5,7 berasal dari step yang diperlukan motor stepper untuk berevolusi penuh yaitu 2048 dibagi dengan  $360^\circ$ .

Untuk pengujian dengan arah mata angin *set-point*  $0^\circ$ , ditunjukkan pada Tabel 4.6. dan Gambar 4.12 dengan arah awal sebesar  $42,72^\circ$ , arah sensor inframerah akan mencapai  $0 \pm 8^\circ$  yaitu  $5,32^\circ$  pada perulangan ke 16. Sedangkan untuk *set-point* 20 dan 40 diperlukan 2 kali perulangan dan *set-point* 60 dengan 1 perulangan seperti yang ditunjukkan pada Tabel 4.8, 4.9, dan 4.10.

**Tabel 4.7.** Pengujian Motor Stepper dengan *Set-Point*  $0^\circ$

Perulangan Ke-	Arah Mata Angin ( $^\circ$ )	Perulangan Ke-	Arah Mata Angin ( $^\circ$ )
0	42,72	9	24,82
1	18,68	10	-39,19
2	-7,77	11	-8,11
3	-22,28	12	32,33
4	23,11	13	27,05
5	-22,44	14	-46,03
6	23,89	15	12,83
7	148,00	16	5,32
8	-21,81		





**Gambar 4.12.** Pengujian Motor Stepper dengan *Set-Point* 0°

**Tabel 4.8.** Pengujian Motor Stepper dengan *Set-Point* 20°

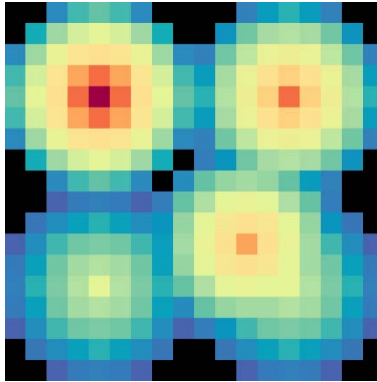
Perulangan Ke-	Arah Mata Angin (°)
0	6,80
1	3,17
2	21,25

**Tabel 4.9.** Pengujian Motor Stepper dengan *Set-Point* 40°

Perulangan Ke-	Arah Mata Angin (°)
0	16,08
1	-7,19
2	46,56

**Tabel 4.10** Pengujian Motor Stepper dengan *Set-Point* 60°

Perulangan Ke-	Arah Mata Angin (°)
0	38,34
1	58,82



**Gambar 4.13.** Pengujian Koordinat Pksel dan Data

## 4.4. Pengujian Perangkat Lunak Pemetaan

### 4.4.1. Pengujian Koordinat Pksel dan Data

Pengujian ini dilakukan untuk menunjukkan koordinat piksel pada gambar dan nilai data pada korrdinat tersebut. Berikut adalah kode program untuk mengujinya:

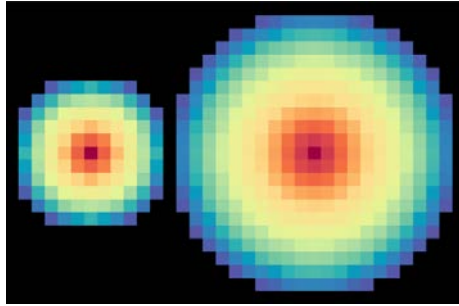
```
static const size_t w = 18, h = 18;
heatmap_t* hm = heatmap_new(w, h);
heatmap_add_weighted_point(hm, 4, 4, 25);
heatmap_add_weighted_point(hm, 13, 4, 20);
heatmap_add_weighted_point(hm, 4, 13, 10);
heatmap_add_weighted_point(hm, 13, 13, 10);
heatmap_add_weighted_point(hm, 13, 13, 10);
```

Ukuran gambar dibuat kecil yaitu 18x18 agar terlihat tiap pikselnya. Pada Gambar 4.13 koordinat (4,4) data bernilai paling tinggi.

### 4.4.2. Pengujian Radius pada Pola

Dalam pemrograman visualisasi ini terdapat pemberian pola terhadap setiap data. Pola ini akan membentuk gradasi nilai data di sekitar koordinat data yang diberikan. Pola ini dapat diatur radiusnya, seperti pada kode program pengujian berikut ini:

```
static const size_t w = 50, h = 30;
heatmap_t* hm = heatmap_new(w, h);
auto pola5 = heatmap_stamp_gen(5);
auto pola10 = heatmap_stamp_gen(10);
heatmap_add_weighted_point_with_stamp(hm, 10, 15, 10, pola5);
heatmap_add_weighted_point_with_stamp(hm, 35, 15, 10, pola10);
```



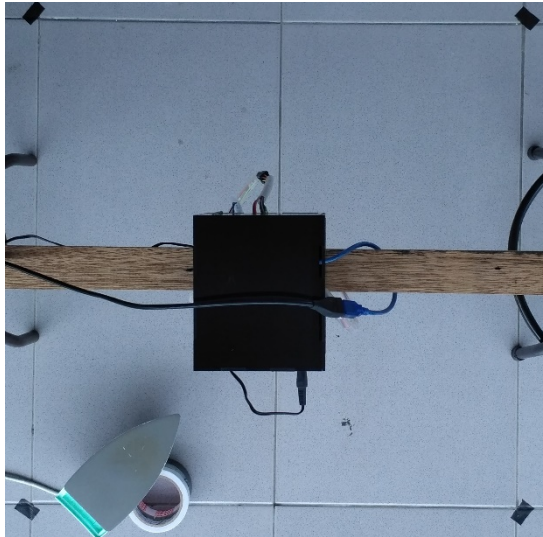
**Gambar 4.14.** Pengujian Radius pada Pola

Dengan data yang bernilai sama yaitu 10, namun dengan radius yang berbeda, menghasilkan gradasi yang sama dan data kanan pada koordinat (35,10) lebih lebar radius polanya yang ditunjukkan pada Gambar 4.14.

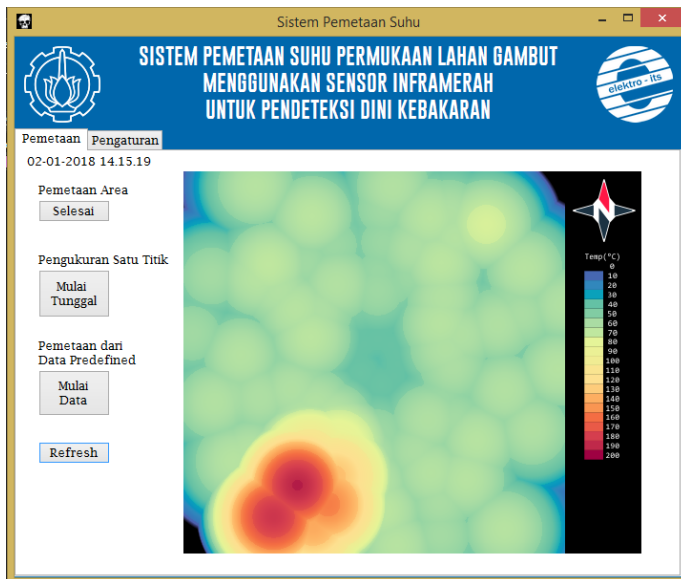
#### 4.5. Pengujian Sistem

Pengujian terhadap sistem secara keseluruhan dilakukan dengan tujuan mengetahui hasil pemetaan suhu suatu area. Pengujian pertama ini dilakukan terhadap area seperti yang ditunjukkan pada Gambar 4.15. dengan terdapat sumber panas berupa setrika listrik yang bersuhu tinggi pada pojok kiri bawah. Sensor inframerah ditempatkan pada ketinggian sekitar 60 cm. Dengan kemiringan sensor maksimum pada program  $45^\circ$ , radius area yang terukur adalah 60 cm dan diameter sebesar 120 cm seperti pada persamaan (3.7). Hasil pengujian ini ditunjukkan pada Gambar 4.16. Dari gambar hasil pemetaan tersebut, diketahui bahwa terdapat objek yang bersuhu tinggi di pojok kiri bawah area. Pada tampilan GUI arah utara ditunjukkan dengan arah atas dengan legenda warna representasi suhu pada sebelah kanan gambar pemetaan.

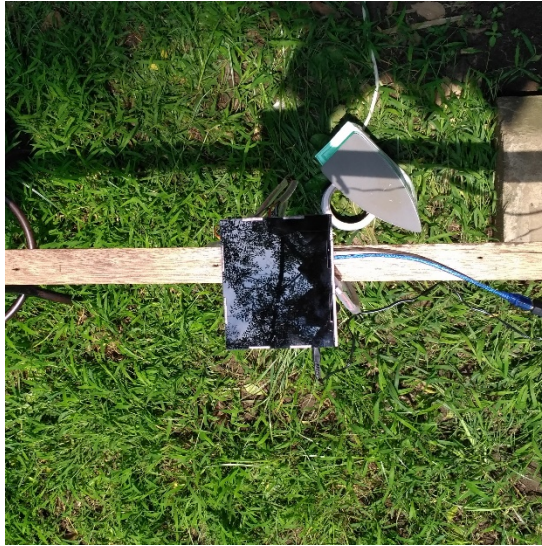
Pengujian ke-2 sistem adalah dengan memetakan suhu lahan berumput seperti pada Gambar 4.17. Seperti pada pengujian sistem yang pertama, sensor inframerah diletakkan pada ketinggian sekitar 60 cm. dengan kemiringan maksimum pada program  $45^\circ$ , radius area yang terukur adalah 60 cm dan diameter sebesar 120 cm. Hasil pengujian ini ditunjukkan pada Gambar 4.18, di mana objek bersuhu tinggi terdapat pada sebelah kanan sistem.



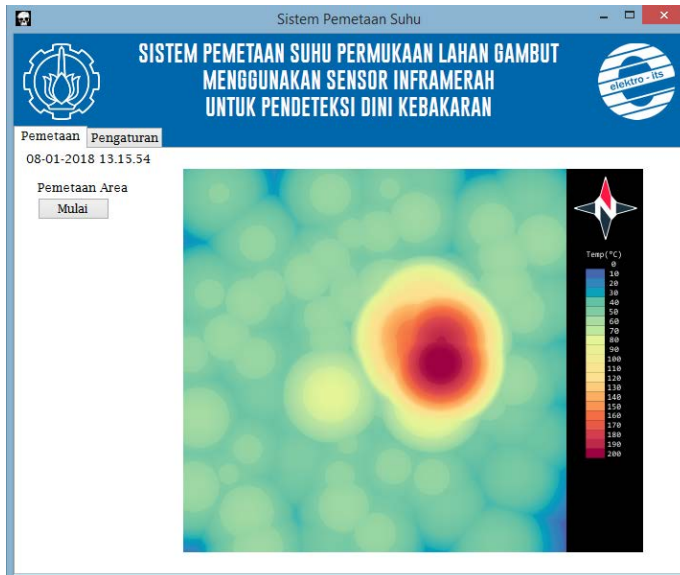
**Gambar 4.15.** Lingkungan Pengujian Sistem



**Gambar 4.16.** Hasil Pengujian Sistem

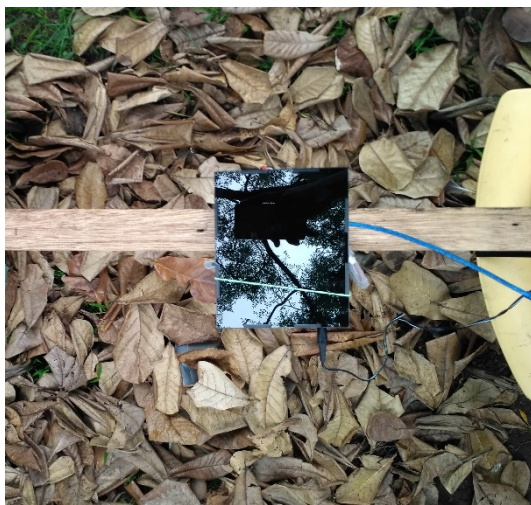


**Gambar 4.17.** Lingkungan Pengujian Ke-2 Sistem

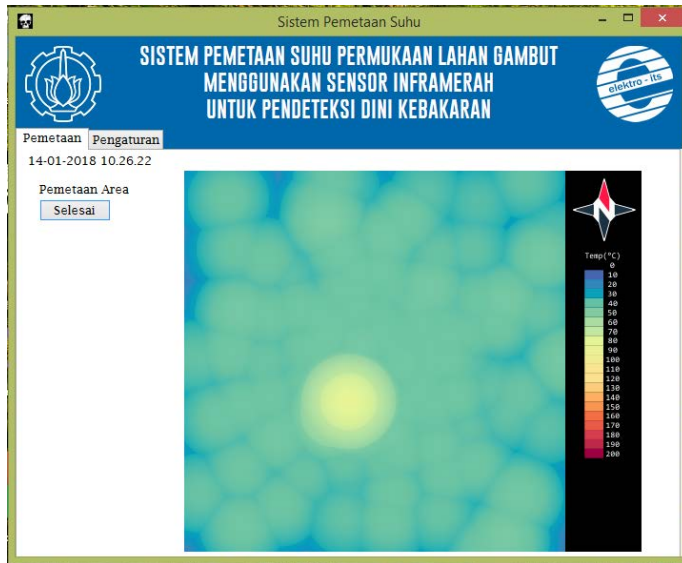


**Gambar 4.18.** Hasil Pengujian Ke-2 Sistem

Pengujian ke-3 sistem adalah dengan memetakan suatu lahan yang didominasi oleh daun kering seperti ditunjukkan pada Gambar 4.19. Pada suatu titik, sebelah kiri bawah sistem, diberikan sumber panas berupa setrika listrik yang diletakkan di bawah dedaunan kering, sehingga permukaan sumber panas hanya sedikit yang terlihat. Pada Gambar 4.19 terlihat sedikit warna abu-abu yang merupakan sumber panas tersebut. Hasil pengujian ini ditunjukkan pada Gambar 4.20 menunjukkan pada sebelah kiri bawah titik tengah GUI terdapat sumber panas dengan suhu sedang direpresentasikan warna kuning.



**Gambar 4.19.** Lingkungan Pengujian Ke-3 Sistem



**Gambar 4.20.** Hasil Pengujian Ke-3 Sistem

*# Halaman ini sengaja dikosongkan #*



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Pada penelitian ini telah dibuat perancangan, realisasi, dan pengujian sistem pemetaan suhu permukaan lahan. Sistem ini terdiri dari sensor inframerah thermopile untuk mengukur suhu objek tanpa kontak. Sistem pemindai menggunakan motor servo dan motor stepper yang dapat mengubah arah titik pengukuran suhu sehingga dapat mencakup seluruh luasan atau area. Mikrokontroler Arduino Nano digunakan untuk membaca sinyal sensor inframerah thermopile dan mengirimnya ke komputer. Hasil pengujian menunjukkan bahwa sistem ini memiliki rerata kesalahan sebesar 6,4% pada rentang pengukuran objek bersuhu 37 °C s.d. 170 °C. Untuk pengukuran terhadap objek bersuhu konstan secara berulang memiliki kesalahan sebesar  $\pm 1,5$  °C. Hasil pengujian sistem menunjukkan bahwa perangkat lunak pemetaan dapat memetakan suhu permukaan lahan dengan suatu titik diberikan sumber panas.

#### **5.2. Saran**

Rangkaian pengondisi sinyal dikembangkan untuk rentang suhu pengukuran yang lebih kecil sesuai kebutuhan sehingga pengukuran dapat lebih akurat / *error* yang lebih kecil. Sistem pemetaan ini digabungkan dengan *wireless sensor network* (WSN) untuk pendeteksi dini kebakaran lahan sehingga mencakup banyak area pengamatan dengan beberapa *node*-nya yang mengandung sistem pemetaan suhu ini.

*# Halaman ini sengaja dikosongkan #*

## DAFTAR PUSTAKA

- [1] Glauber, Ann Jeannette., Moyer, Sarah., Adriani, Magda., dan Gunawan, Iwan. (2016). *Kerugian dari Kebakaran Hutan – Analisa Dampak Ekonomi dari Krisis Kebakaran Tahun 2015*. Jakarta: World Bank Group. pp2.
- [2] Wetlands International Indonesia. (2017). *Apa lahan basah itu? - Wetlands International Indonesia*. [online] Available at: <https://indonesia.wetlands.org/id/wetlands/apa-lahan-basah-itu/> [Accessed 9 Feb. 2017].
- [3] Akbar, Luthfan Aufar., Muhammad Rivai, dan Budiman, Fajar. (2016). Rancang Bangun Sensor Node pada Wireless Sensor Network Menggunakan Deret Sensor Gas dan Jaringan Syaraf Tiruan untuk Mendeteksi Kebakaran Hutan. *Jurnal Teknik ITS*, 5(2), pp.A192 s.d. A197.
- [4] S, P. (2017). *Mengapa kebakaran lahan gambut sulit dipadamkan*. [online] <https://beritagar.id/>. Available at: <https://beritagar.id/artikel/sains-tekho/mengapa-kebakaran-lahan-gambut-sulit-dipadamkan> [Accessed 9 Feb. 2017].
- [5] Roswintiarti, Orbita. (2016). *Informasi Titik Panas (Hotspot) Kebakaran Hutan/Lahan*. Jakarta: Lapan.
- [6] Dwinata, Irwan Candra. Muhammad Rivai, dan Setijadi, Eko. (2016). Desain Wireless Sensor Network dan Webserver untuk Pemetaan Titik Api pada Kasus Kebakaran Hutan. *Jurnal Teknik ITS*, 5(2), pp.A198 s.d. A203.
- [7] Robel, Jeffrey. dan Graumann, Axel. (2014). *NOAA KLM User's Guide*. Asheville: NOAA, pp2-15.
- [8] Giglio, Louis., Schroeder, Wilfrid., Dan Justice, Christoper O. (2016). The Collection 6 MODIS Active Fire Detection Algorithm and Fire Products. *Remote Sensing of Environment*, 178, pp. 31-41
- [9] TheFreeDictionary.com. (2017). *Peaty soil*. [online] Available at: <http://encyclopedia2.thefreedictionary.com/Peaty+soil> [Accessed 9 Feb. 2017].
- [10] Fahmi, A. dan Risnandar, C. (2017). *Lahan gambut - Ensiklopedi Jurnal Bumi*. [online] Jurnal Bumi. Available at: <https://jurnalbumi.com/lahan-gambut/> [Accessed 9 Feb. 2017].
- [11] Gruner, Klaus-Dieter. (2003). *Principles of Non-Contact Temperature Measurement*. California: Raytek, 7.

- [12] Omega.com. (2017). *Infrared Thermometer*. [online] Available at: <http://www.omega.com/prodinfo/infraredthermometer.html> [Accessed 9 Feb. 2017].
- [13] Tsai, CF. dan Young, MS. (2003). Pyroelectric Infrared Sensor-Based Thermometer for Monitoring Indoor Objects. *America Institute of Physics: Review of Scientific Instruments*, 74(12), pp. 5267-5273.
- [14] Webster, J.G. dan Eren, Halit. (2014). *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*. 2th Ed. Boca Raton Florida: CRC Press, pp72-10 s.d. 72-14.
- [15] Chen, Shung-Nan. (2011). Temperature Error Analysis and Parameter Extraction of an 8-14  $\mu\text{m}$  Thermopile with a Wavelength-Independent Absorber for Tympanic Thermometer. *IEEE Sensors Journal*, 11(10), pp. 2310-2317.
- [16] Garcia Saez, A., Quero, J. and Angulo Jerez, M. (2016). Earth Sensor Based on Thermopile Detectors for Satellite Attitude Determination. *IEEE Sensors Journal*, 16(8), pp.2260-2271.
- [17] Fraden, Jacob. (2016). *Handbook of Modern Sensors – Physics, Design, and Application*. 5<sup>th</sup> Ed. London: Springer, pp. 599 s.d. 607.
- [18] Webster, J.G. dan Eren, Halit. (2014). *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*. 2th Ed. Boca Raton Florida: CRC Press, pp69-5 s.d. 69-7.
- [19] Munna, MS., Bhuyan, MAI., Rahman, KM., and Hoque, MA. (2015). Design, Implementation and Performance Analysis of a Dual-Axis Autonomous Solar Tracker. *3<sup>rd</sup> International Conference on Green Energy and Technology*.
- [20] Tarnini, MY. (2015). Fast and Cheap Stepper Motor Drive. *4<sup>th</sup> International Conference on Renewable Energy Research and Applications*, pp. 689-693.
- [21] Basic Electronics Tutorials. (2017). *Operational Amplifier Basics - Op-amp tutorial*. [online] Available at: [http://www.electronics-tutorials.ws/opamp/opamp\\_1.html](http://www.electronics-tutorials.ws/opamp/opamp_1.html) [Accessed 9 Feb. 2017].
- [22] Basic Electronics Tutorials. (2017). *Non-inverting Operational Amplifier - The Non-inverting Op-amp*. [online] Available at: [http://www.electronics-tutorials.ws/opamp/opamp\\_3.html](http://www.electronics-tutorials.ws/opamp/opamp_3.html) [Accessed 9 Feb. 2017].

- [23] Basic Electronics Tutorials. (2017). Differential Amplifier - The Voltage Subtractor. [online] Available at: [http://www.electronicstutorials.ws/opamp/opamp\\_5.html](http://www.electronicstutorials.ws/opamp/opamp_5.html) [Accessed 9 Feb. 2017].
- [24] Basic Electronics Tutorials. (2017). *DC Motors and Stepper Motors used as Actuators*. [online] Available at: [http://www.electronicstutorials.ws/io/io\\_7.html](http://www.electronicstutorials.ws/io/io_7.html) [Accessed 9 Feb. 2017].
- [25] Jameco.com. (2017). *How Servo Motors Work*. [online] Available at: <https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html> [Accessed 9 Feb. 2017].
- [26] Robotpark.com. (2017). *Stepper Motor Working*. [online] Available at: <http://www.robotpark.com/Stepper-Motor-Working> [Accessed 9 Feb. 2017].
- [27] NASA. (2017). *Representation of Earth's Invisible Magnetic Field*. [online] Available at: [https://www.nasa.gov/mission\\_pages/sunearth/news/gallery/Earths-magneticfieldlines-dipole.html](https://www.nasa.gov/mission_pages/sunearth/news/gallery/Earths-magneticfieldlines-dipole.html) [Accessed 9 Feb. 2017].
- [28] Pcb.com. (2017). *Introduction to MEMS Accelerometers*. [online] Available at: <http://www.pcb.com/Resources/Technical-Information/mems-accelerometers> [Accessed 9 Feb. 2017].
- [29] Agarwal, T. (2017). *PIC Microcontroller Architecture with Applications*. [online] Available at: <https://www.efxkits.co.uk/pic-microcontroller-architecture-with-applications/> [Accessed 9 Feb. 2017].

*# Halaman ini sengaja dikosongkan #*

## LAMPIRAN

### A. Program Pada Arduino

```
#include <Servo.h>
#include <Stepper.h>
#include <Wire.h>
Servo myServo;
String buff, delim=",";
char myChar = 0;
int i=0, a=0, b=0;
int kondisi = 99;
const int tpPin = A3;
const int trPin = A2;
float suhuObjek;
float suhuObjekMean;

void setup() {
  Serial.begin(9600);
  myServo.attach(9);
  Wire.begin();
  kondisi = 1;
  writeI2c(accelAddress, accelSleepMgmt, 0x00);
  writeI2c(magnAddress, magnMode, 0x00);
  putarStepper(90);
}
//=====================================================
void loop() {
  switch(kondisi){
    case 1:
      terimaM();
      break;
    case 2:
      a=0; b=0;
      kirimDataSensor();
      terimaS();
      break;
  }
}
//=====================================================
void getSuhu(){
  float tpADC, tpVout, tpVoff=2.15, tpVtp;
  float trADC, trVout;
  //persamaan thermistor modelsteihart-hart
  float trA = 788.759E-6, trB = 206.32E-6, trC = 125.448E-9;
  //rangkaian pengondisi sinyal thermistor
  float trVcc = 5, trVoff = 2.46, trR8 = 22000, trR9 = 22000,
```

```

    trR10 = 33000;
    float trX, trRTr;
    float trSuhu;
    delay(25);
    trADC = analogRead(trPin);
    tpADC = analogRead(tpPin);
    trVout = trADC*5/1024;
    trX = ((trVout-
        trVoff)/trVcc)+(trR10/trR8))*(trR8/(trR8+trR10));
    trRTr = (trX*trR9)/(1-trX);
    trSuhu = 1/(trA+trB*log(trRTr)+trC*pow(log(trRTr),3));
    tpVout = tpADC*5/1024;
    tpVtp = (tpVout-tpVoff)/10000;
    suhuObjek = trSuhu + tpVtp/(4*2.65E-15*pow(trSuhu,3))-273.15;
}
//=====
void kirimDataSensor(){
    for (a=0; a<9; a++){
        //atur servo
        putarServo(85-(5*a)); //85 pada a=0, 45 pada a=8
        for (b=0; b<18; b++){
            //atur stepper
            aturStepper();
            suhuObjekMean=0;
            for (i=0; i<10; i++){
                getAccel();
                getMagn();
                getSuhu();
                suhuObjekMean+= 0.1*suhuObjek;
                delay(25);
            }
            buff =
            a+delim+b+delim+tiltFild+delim+magnHeadFild+delim+suhuObjekMean
            ;
            Serial.println(buff);
            delay(50);
            terimaS();
        }
        //dari 340 kebalik ke 180 dulu, agar kabel tidak terlilit
        b=9;aturStepper();
    }
}

void aturStepper(){
    int stepperError;
    int stepperCurrent;
    int stepperSetPoint=b*20;
    int magnHeadTop = stepperSetPoint + 8;
    int magnHeadBottom = stepperSetPoint - 8;

```



```

for (i=0; i<10; i++){
    getMagn();
    delay(50);
}
if (stepperSetPoint==0){
    if (magnHeadFild>350)magnHeadFild=0;
}

while((magnHeadTop<magnHeadFild) ||
(magnHeadBottom>magnHeadFild)){
    stepperCurrent = magnHeadFild;
    if ((stepperSetPoint>225) &&
(stepperCurrent<90))stepperCurrent+=360;
    if ((stepperSetPoint<90) &&
(stepperCurrent>225))stepperCurrent-=360;
    stepperError=stepperSetPoint-stepperCurrent;
    putarStepper(-(stepperError*5.7));
    for (i=0; i<10; i++){
        getMagn();
        delay(50);
    }
    if (stepperSetPoint==0){
        if (magnHeadFild>350)magnHeadFild=0;
    }
}
}

```

## B. Program Pada Microsoft Visual Studio

```
int kondisi = 99; //state-machine 99,0,1,2,3,4
String^ delimStr = " ,\r\n";
const size_t rStamp = 80, d = 2 * rStamp + 1, w = 400, h = 400;
int jumRadius = 9, jumSpot = 18; //9*18=162
int miringMax = 45; //step 5 derajat dari 5 derajat
int* miring = (int *)calloc(162, sizeof(int));
int* kompas = (int *)calloc(162, sizeof(int));
int* suhu = (int *)calloc(162, sizeof(int));
String^ terimaSerialData;
String^ terimaSerialCon;
array<Char>^ delimiter;
array<String>^ serialDatas;
int dataCacahMiring, dataCacahSudut;
int cacahSpot, cacahRadius, pikselStepMiring, pikselRadius, x, y;
float weight, dataMiring, dataSudut, dataSuhu;
double pikselStepSudut;

private: System::Void serialPort1_DataReceived(System::Object^
sender, System::IO::Ports::SerialDataReceivedEventArgs^ e)
{
    if (kondisi == 2 || kondisi == 3 || kondisi == 4) {
        terimaSerialData = serialPort1->ReadLine();
        delimiter = delimStr->ToCharArray();
        serialDatas = terimaSerialData->Split(delimiter);
        if (serialDatas->Length > 4) {
            dataCacahMiring = (int)(Convert::ToInt16(serialDatas[0]));
            dataCacahSudut = (int)(Convert::ToInt16(serialDatas[1]));
            dataMiring = (float)(Convert::ToDouble(serialDatas[2]));
            dataSudut = (float)(Convert::ToDouble(serialDatas[3]));
            dataSuhu = (float)(Convert::ToDouble(serialDatas[4]));
            *(miring + jumSpot*dataCacahMiring + dataCacahSudut) = dataMiring;
            *(kompas + jumSpot*dataCacahMiring + dataCacahSudut) = dataSudut;
            *(suhu + jumSpot*dataCacahMiring + dataCacahSudut) = dataSuhu;
        }
    }
}

private: System::Void buttonGetPort_Click(System::Object^ sender,
System::EventArgs^ e) {
    array<String>^ serialPorts = nullptr;
    serialPorts = serialPort1->GetPortNames();
    for each(String^ port in serialPorts)
    {
        comboBoxPortNames->Items->Add(port);
    }
}
```

```

private: System::Void
comboBoxPortNames_SelectedIndexChanged(System::Object^ sender,
System::EventArgs^ e) {
Object^ selectedItem = comboBoxPortNames->SelectedItem;
serialPort1->PortName = selectedItem->ToString();
kondisi = 0;
}
private: System::Void buttonConnect_Click(System::Object^ sender,
System::EventArgs^ e) {
if (kondisi == 0)
{
kondisi = 1;
serialPort1->Open();
serialPort1->WriteLine("s");
buttonConnect->Text = "Disconnect";
}
else if (kondisi == 1)
{
kondisi = 0;
serialPort1->Close();
buttonConnect->Text = "Connect";
}
}

private: System::Void buttonMulai_Click(System::Object^ sender,
System::EventArgs^ e) {
switch (kondisi)
{
case 0:
//perintah untuk menyambungkan ke serial
break;
case 1:
//perubahan kondisi dari 1 menjadi 2;
kondisi = 2;
//kirim char ke arduino untuk mulai kerja
serialPort1->WriteLine("m");
buttonMulai->Text = "Selesai";
break;
case 2:
kondisi = 1;
//kirim char ke arduino untuk selesai kerja
serialPort1->WriteLine("s");
buttonMulai->Text = "Mulai";
break;
default:
break;
}
}
}

```

```

private: System::Void Refresh_Click(System::Object^ sender,
System::EventArgs^ e) {
if (kondisi == 2 || kondisi == 3 || kondisi == 4) {
label6->Text = terimaSerialData;
if (serialDatas->Length > 2) {
label7->Text = serialDatas[0];
label8->Text = serialDatas[1];
label9->Text = serialDatas[2];
}
//inisialisasi
heatmap_t* hm = heatmap_new(w, h);
//visualisasi
//auto pola = heatmap_stamp_gen(rStamp);
auto pola = heatmap_stamp_gen_nonlinear(rStamp);
for (int cacahRadius = 0; cacahRadius < jumRadius; cacahRadius++)
{
for (int cacahSpot = 0; cacahSpot < jumSpot; cacahSpot++) {
weight = *(suhu + jumSpot*cacahRadius + cacahSpot);
pikselStepMiring = *(miring + jumSpot*cacahRadius + cacahSpot);
pikselStepSudut = *(kompas + jumSpot*cacahRadius + cacahSpot);

pikselRadius = ((w / 2) - (rStamp /
4))*(tan(pikselStepMiring*PI/180) / tan(miringMax*PI / 180));

x = w / 2 + pikselRadius*sin(pikselStepSudut*PI / 180);
y = h / 2 - pikselRadius*cos(pikselStepSudut*PI / 180);

heatmap_add_weighted_point_with_stamp(hm, x, y, weight, pola);
}
}
std::vector<unsigned char> image(w*h * 4);
heatmap_render_default_to(hm, &image[0]);
heatmap_free(hm);
if (unsigned error = lodepng::encode("heatmap.png", image, w, h))
{
std::cerr << "encoder error " << error << ": " <<
lodepng_error_text(error) << std::endl;
}
pictureHeatmap->ImageLocation = "heatmap.png";

DateTime localDate = DateTime::Now;
String^ localDateStr = localDate.ToString("dd'-'MM'-'yyyy
HH'.'mm'.'ss");
labelTanggalWaktu->Text = localDateStr;
}
}

```

## **BIODATA PENULIS**



Penulis dilahirkan di Yogyakarta pada tanggal 16 Januari 1994. Sebagai anak kedua dari tiga bersaudara, penulis mengawali kegiatan pendidikan formal pada tahun 2000 di SD Negeri Tamansari 2 Yogyakarta, yang kemudian dilanjutkan di SMP Negeri 2 Yogyakarta, SMA Negeri 2 Yogyakarta dan pada tahun 2012 penulis diterima sebagai mahasiswa di jurusan Diploma Teknik Elektro Sekolah Vokasi UGM Yogyakarta. Pada tahun 2015, penulis melanjutkan studi di program sarjana lintas jalur Teknik Elektro ITS Surabaya.

Email :

widya.yanuar.s.0116@gmail.com